POWER – AWARE RESOURCE MANAGEMENT

FOR

SENSOR – ACTOR NETWORKS

By

ABHISHEK GHALE

Bachelor of Science in Computer Engineering

Pokhara University

Pokhara, Nepal

2004

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 2009

POWER – AWARE RESOURCE MANAGEMENT

FOR

SENSOR – ACTOR NETWORKS

Thesis Approved:

Dr. Xiaolin (Andy) Li

Thesis Adviser

Dr. Douglas R. Heisterkamp

Dr. Johnson P Thomas

Dr. A. Gordon Emslie

Dean of the Graduate College

ACKNOWLEDGMENTS

The initial idea of research in “Data Pipelining for Sensor Actuator Network” came during the discussion with Dr. Xiaolin Li. Later, I found the idea of implementation of data pipelining helps save power and resources for sensor actuator network. I would like to express my gratitude to Dr. Xiaolin Li for his guidance and support without which, this would not be possible.

I am grateful to Dr. Douglas R. Heisterkamp and Dr. Johnson P Thomas for willing to be in my thesis committee and helping me with questions and suggestions. I would like to thank all the professors from the computer science department who have contributed to this work directly and indirectly.

I am very thankful for having such a caring and supporting family. Without their guidance and support none of this would have been possible. I would also like to thank Laura Gann and her staff of Engineering Distance at OSU for never ending support.

Last, but not the least I would like to thank my friends Binod Gurung, Pradeep Baral, Prakash Giri, Nabin Ghimire, and Milan Kattel for being there for me all the time

TABLE OF CONTENTS

Chapter

I. INTRODUCTION 1

II. REVIEW OF LITERATURE...4

- 2.1 Mobile Sensor Actuator Network 4
- 2.2 Sink Mobility 5
- 2.3 Delay Tolerant Networks...6
- 2.4 Relay Sensor Node..8
- 2.5 Message Ferry...9
- 2.6 Data Pipelining..12

III. PROBLEM STATEMENT 13

IV. METHODOLOGY, MODELS AND EXPERIMENTAL RESULTS 15

- 4.1 Methodology 15
- 4.2 Energy Models 19
- 4.3 Algorithm generating deployment and data pipelining 20
- 4.4 Experimental results...24

V. CONCLUSION 36

REFERENCES 38

APPENDICES 43

LIST OF TABLES

Table Page

4.4.1 Time to deliver – varying number of mobile nodes..29

4.4.2 Energy required to delivery packets – varying number of mobile nodes 30

4.4.3 Number of packets delivered and energy consumed in 300 Unit time............32

4.4.4 Life time & number of packets deliver with constant energy for mobile nodes 34

LIST OF FIGURES

Figure Page

2.1 Bull wearing collar containing FLECKTM 4

2.2 Diagram of the sensor network with mobility sink 6

2.3 A Physical cyclic Mobi Space 7

2.4 Relay node in a static network with fixed flow 9

2.5 Message Ferry Model 10

2.6 56 Node on main span of Golden Gate Bridge..11

2.7 Transfer time for n-hop network using pipelining 12

3.1 Mobile actuator node transferring data to base station 13

4.1.1 Data transfer between sink node and base station 16

4.1.2 Data traveling in pipeline tunnel 16

4.1.3 Design of mobile sensor actors in pipelining environment 17

4.3.1 Pipelining for even number of mobile nodes..23

4.3.2 Pipelining for odd number of mobile nodes 24

4.4.1 GUI for simulator 25

4.4.2 Traditional mobility pattern of mobile nodes 27

4.4.3 Pipelining mobility pattern of mobile nodes 28

4.4.4 Time to deliver packets – varying number of mobile nodes 29

4.4.5 Energy require to deliver packets – varying number of mobile nodes 31

Figure Page

4.4.6 Number of packets delivered in 300 Unit time 32

4.4.7 Energy consumed by mobile nodes in 300 Unit time 33

4.4.8 Life time of mobile nodes 34

CHAPTER I

INTRODUCTION

Rapid advances in wireless communication, processor and miniature electronic devices have made pervasive sensing possible through sensor networks. A sensor network consists of low cost sensor nodes that are equipped with radio communication and powered by battery. Sensor networks have deployed for a wide spectrum of application in science, engineering and military. For example, wildlife reservation, healthcare, precision agriculture [12], border protection, battles field and hostile environment [13], emergency responses and various other applications. Sensor nodes are used to collect data in sensor network. Sensor nodes are smart and can easily be deployed.

In a wireless sensor network, a sensor node collects and transfers data to base station. A sensor node uses its neighboring sensor nodes to transfer data to different base stations by organizing themselves; forming a node to node multi hop data propagation network [1, 14].

As sensor nodes are battery driven, replacing the battery is infeasible in large network therefore; energy is an important limiting factor for sensor network. These tiny devices have low processing power compared to larger systems and the amount of energy required increases with distance. Therefore, one of the most challenging issues in wireless sensor networks is to increase efficiency in terms of energy consumption.

Sensor-actuator networks can be used to conserve the energy of sensor nodes. Sensor-actuator network have sensor nodes that can sense the environment then transmit data to nodes that can move and communicate among themselves e.g. mobile robots, robotic arms etc. Actuator nodes use the wireless medium to communicate and perform distributed sensing by moving to different locations. Actuator nodes are resource rich, capable of processing information better, making decisions and performing required tasks. Sensor actuator node operates autonomously in an unattended environment such as battlefield, area of explosion, fire etc.

Message-ferrying has been in common practice for collecting and delivering the data in sensor-actor network. Message-ferry is capable of moving within a network, collecting data from one sensor node and delivering the data to base station. Such mobile nodes are capable of travelling long distances in places where end to end connectivity is not possible [9].

To conserve the mobile nodes energy, concept of pipelining is used where transfer data takes place with an increased speed. The purpose of this research is to show that efficient management of mobile nodes via pipelining will not only conserve power but also help to increase the lifetime of a sensor actuator network.

CHAPTER II

REVIEW OF LITERATURE

2.1 Mobile Sensor Actuator Network

Autonomous animal control is an application of mobile sensor actuator network. Cattle breeding industries suffer when a fight occurs between bulls in a farm. To avoid this loss, dynamic animal state estimation, real time actuation and efficient mobile wireless transmission are required. An actuation device is placed on the neck of each bull, as shown in Fig [2.1]. Attached to this device is a GPS sensor that keeps track of position, speed and direction to which a bull is heading. When two bulls come together, the device is activated to separate the bulls from each other. This information can be exchanged in the network via the nodes and stored for future references [10].

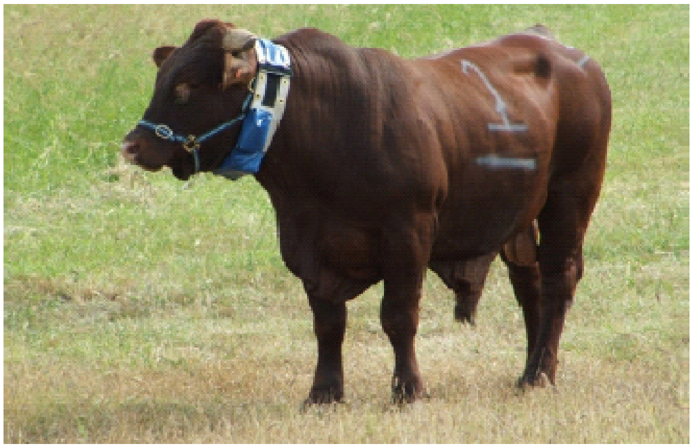


Fig 2.1: Bull wearing collar containing FleckTM hardware, batteries and antennas.

Courtesy: Liu, C and J. Wu [10].

2.2 Sink Mobility

Sink mobility increases life of network by more than 50% and reduces energy dissipation by 30% [1]. This can be achieved with the help of four mobility patterns. First proposed pattern includes random walk and passive data collection where data is passively cached in sensor nodes and beacon message is periodically transmitted. In second pattern, mobile sink node visits maximum sensor nodes in a single move and travels less distance. Biased random walk with passive data collection is another pattern where area of node visit depends on two factors; frequency and population density of sensor nodes within a given area. Deterministic walk with multi–hop data propagation is another pattern where a mobile sink node moves on a predefined trajectory, line or circle in the network. This pattern is more expensive than others but delivers data with low latency [1].

In predictable sink mobility model, a sink node is placed on a public transportation vehicle which traverses wide sensor network area at once [3], as in Fig [2.2]. Sink nodes broadcast a beacon message continuously and data is retrieved from node by waking up sensor nodes. For the first cycle, sensor nodes observe pattern and range of sink movement. In the second cycle, a collision – resolution based on 802.11 CSMA/ CA is used to transmit data packets when nodes are within range. Sink nodes use communication priority when multiple nodes are within the range and higher priority is given to sensor nodes that are to be out of range. Sink sends wake up signal to sensor nodes saving incredible amount of energy. Main drawback of this model is that sink travel the same path.

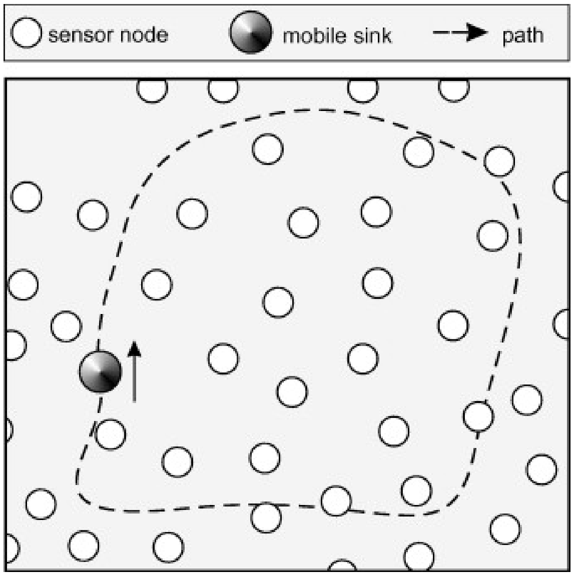


Fig 2.2: Diagram of the sensor network with mobile sink. Courtesy: Pazzi, R.W.N. and A. Boukerche [5].

Messages can be relayed – stored temporarily at moving node and waits for opportunities to forward [4]. This scenario can be implemented in vehicles on highways. Delivery of messages improves when movement of vehicles with opposing direction on bidirectional highway and multilane traffic with same direction increases. Low vehicle densities scenario cannot support the delay sensitive application [4].

2.3 Delay Tolerant Networks

Delay Tolerant Networks are occasionally connected and frequently partition networks. This is due to time varying nature of network. Routing of actor node with full knowledge outperforms routing with zero knowledge although there might be the case where recourses are limited. Networks that take smart algorithm approach such as earliest delivery with local query, earliest delivery with all queries perform well in these environments [2].

In [6], proposed routing protocol called routing in Cyclic Mobi Space uses least expected minimum delay. The network models a probabilistic time space graph which saves information and prior knowledge. The time space graph can be translated into probabilistic state – space graph. Expected minimum delay was achieved using Markov decision. In cyclic mobi space routing protocol, mobile nodes have common cyclic motion with contact pattern. For example, nodes moving in a line, circle or rectangle and nodes meet each other at some point, as in Fig [2.3]. This routing protocol outperforms the traditional protocol in terms of delivery rate and delay.

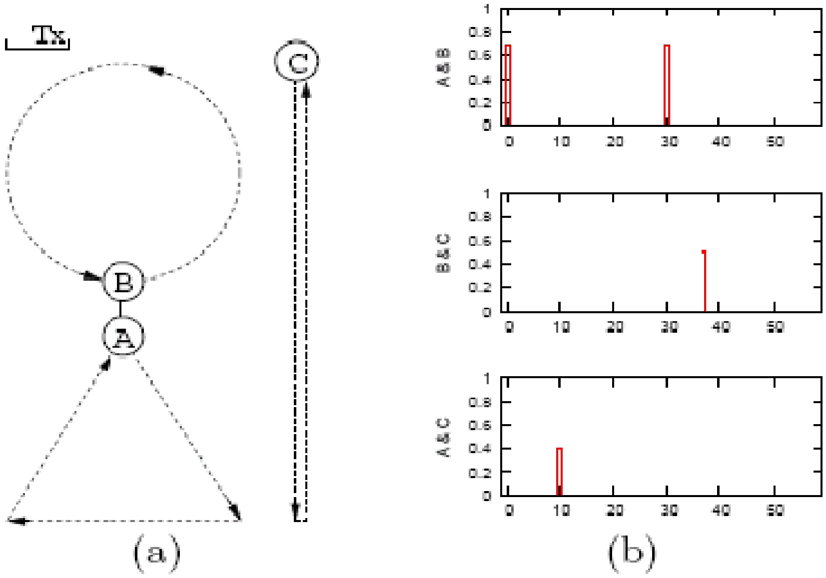
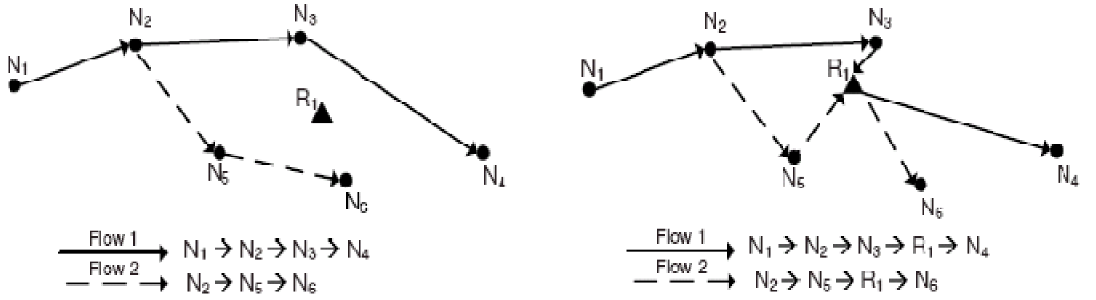


Fig 2.3: A physical cyclic MobiSpace (a). The discretized probabilistic contacts between each pair of nodes of a common motion cycle T (b). Courtesy: Liu, C. and J. Wu [6].

Wearable computers with packet transport mechanism are good option in highly partitioned Adhoc Network [7]. This decreases size and cost of these systems and allows agent to move and communicate. Mobile computer can carry data packets across partitioned network. “Drop Least Encountered Visited” strategy can be implemented for least visited sensor nodes. Adaptive drop strategy algorithm helps to pass packets to destination and partition node is visited and increases bandwidth of the network [7].

2.4 Relay Sensor Node

Deployment of relay nodes with optimal positioning and minimizing overall power is very challenging area of research. In [11], proposed a mobile prediction base relay deployment framework. Paper discusses deployment of relay nodes both in static and mobile network. Power required to transmit is always proportional to the distance between sender and a receiver. Energy consumed depends on position of traditional nodes and relay nodes. In Fig [2.4] (a), nodes can observe multiple paths from N1 to N4. This process consumes lots of energy, using relay node as in Fig [2.4] (b) and helps reduce energy consumption as distance to travel is comparatively less than [2.4] (a). Service set of each relay nodes was computed that includes data rate and distance. Optimal position of relay nodes was chosen according to minimum service set. In case of mobility of nodes, set of flow of network was assumed not to be changed for fixed time duration called epochs. As flow of network changes periodically and nodes are mobile, information need to be collected for every epoch. As relay nodes is assumed to be energy rich, calculation of service set will not affect the traditional nodes.



(a) Data flows without using a relay node (b) Data flows after deploying the relay node

Fig 2.4: Relay node in a static network with fixed flows. Courtesy: Aravindhan, V., S. Venkatesh [11].

2.5 Message Ferry

Message Ferry approach addresses problem of Mobile Adhoc network when network partition is common. In message ferry, movement of ferry remains no randomness and helps deliver data safely. [9] Explored variation of proactive movement of ferry nodes and proved that message ferry is efficient in data delivery and energy consumption. Two schemes that paper discusses are: 1) Node initiated message ferry scheme where ferry moves in the predefined route and sensor nodes knows the ferry route. Also, mobile nodes make a proactive movement to meet ferry and broadcast message to ferry. 2) Ferry initiated message scheme where ferry makes proactive movement to meet sensor nodes and exchange data. Message ferry scheme delivers more messages with less energy [9].

[8], explored challenges in designing message based ferry and purposes a message ferry route algorithm called optimized way point and waiting time at each way points. This algorithm does not require collaboration between sensor nodes and ferry. Sets of way points like station are chosen based on mobility of nodes. Ferry contacts every node with certain minimum probability and determines end to end message delivery and frequency

of nodes ferry meeting. Ferry travels in same path repetitively until nodes can deliver and receive data as in Fig [2.5]. Choosing appropriate way point and corresponding waiting time is an issue in designing ferry route algorithm. Ferry remains in way point until it meets required node or until time is expired or until ferry proceeds. Another issue is constructing path for ferry. Choosing minimum route is similar to a traveling salesman's problem. Making long route may minimize waiting time for ferry in way point but increase number of way points. Reducing way points will eventually increase wait time since nodes need to visit way point. A heuristic solution has been chosen because way points are set closer to each other and making it shorter route. If nodes are spread in large area, then way point are set closer to center of a region. Optimized way point performs better as choosing traveling time and waiting time increases contact probability with nodes [8].

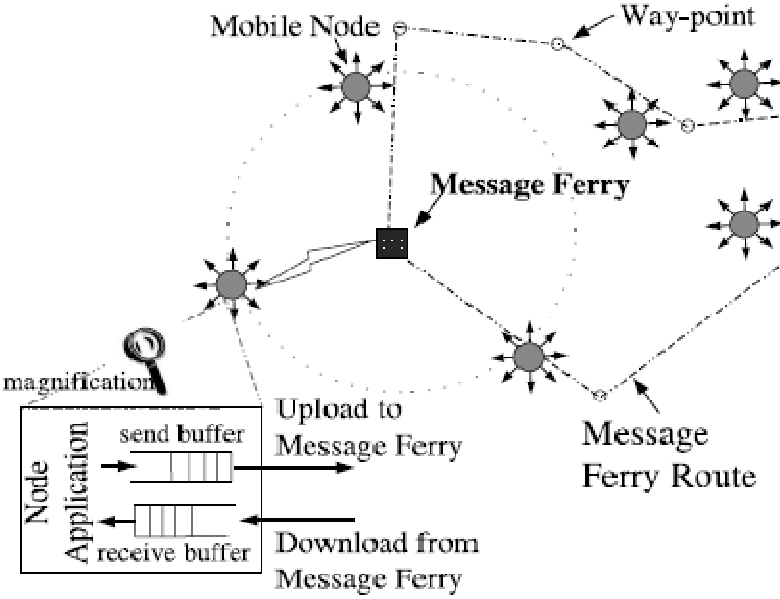


Fig 2.5: Message Ferrying Model. Courtesy: Muhammad Bin, T., A. Mostafa[8].

Wireless Sensor Network is in use to monitor structural health of big structure such as Golden Gate Bridge Fig 2.6. Sensor nodes were designed to monitor vibration and then communicate it with neighboring nodes in the structure. These nodes were equipped with four accelerometer, microcontroller and wireless communication in multi - hop network. TinyOS Operating System was implemented that provided reliable components for data pipelining and high frequency sampling. The prototype was deployed in the long span bridge with 64 nodes of Golden Gate Bridge [17]. MicaZ mote was selected as sensor nodes, with 512kB flash memory that could store 250,000 2 – byte data samples and 2.4 GHz radio frequency (RF) chipcon CC2420 transceiver that supported bi-directional antenna to achieve higher efficiency. MicaZ also consumes less power compared to its class of mote like iMote. Micaz consumes 24mW in active mode compared to 139.5 mW for iMote2 [17].

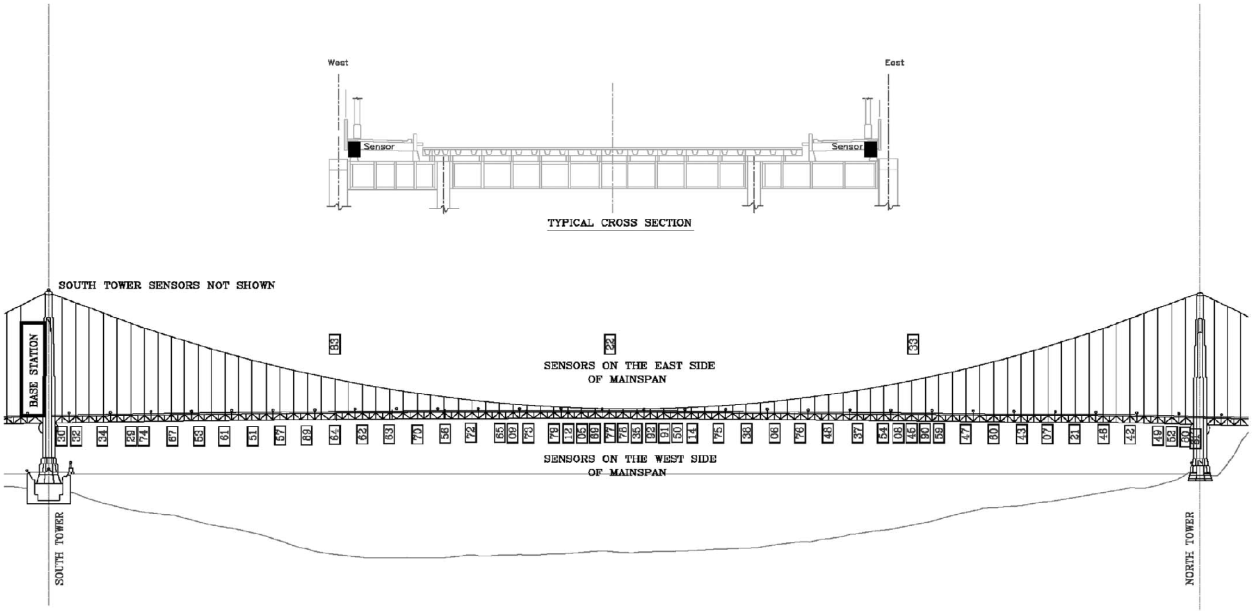


Fig 2.6: 56 Node on main span of Golden Gate Bridge. Courtesy: Ahamim N. Pakzad, Gregory L. Fenves, Sukaun Kim, and David E. Culler [17].

2.6 Data Pipelining

Concept of data pipelining was primarily focuses on data communication among sensor nodes. Sensor nodes behave as intermediate nodes and forward message to another nodes and finally to base station. This helps in reusing of network bandwidth resulting increase throughput of network. If K is the number of hops between transmitting nodes, n is the number of hops in route, T is transfer time of one packet – then, without pipelining, it takes mnT time to transfer m packets. Pipelining with length K results in (n+mk-K)T time [17].

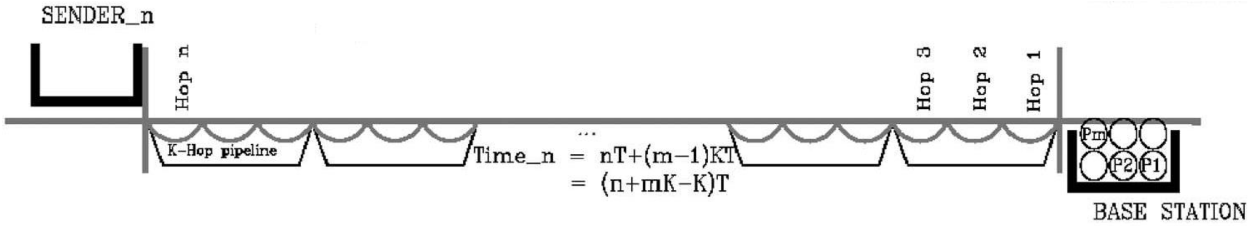


Fig 2.7. Transfer time for n-hop network using pipelining. Courtesy: Ahamim N. Pakzad, Gregory L. Fenves, Sukaun Kim, and David E. Culler [17].

CHAPTER III

PROBLEM STATEMENT

There are many areas where it is hazardous for humans to go and retrieve data. In such situations, mobile actuator nodes are used to retrieve data. It is one of the few efficient ways to deliver packets to base stations. Fig 3.1 shows a mobile actuator node used to retrieve data and deliver data packets to base station. Nuclear and chemical disaster areas, rough terrain, war zones are some of the examples where this technology can be useful.

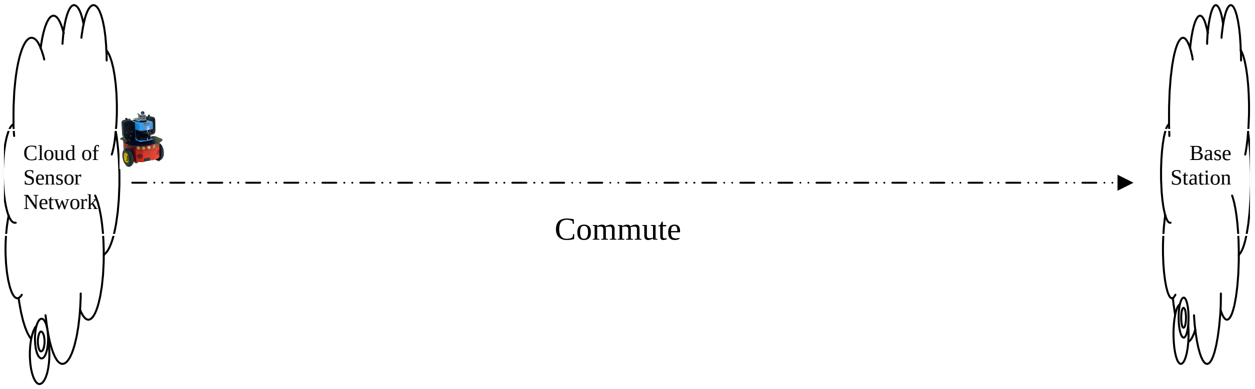


Fig 3.1: Mobile actuator node transferring data to base station

Delivery time and power management have always been an issue for sensor actuator network. Efficient and quick management of sensor actuators can help increase throughput and lifetime of sensor network. Mobile actuator nodes are used to transfer data from sensor network to base station. Robots are commonly used as mobile actuator nodes. These mobile nodes should have sufficient amount of energy and helps conserve power of sensor network.

Conventionally, each mobile node receives set of data and travels required distance to deliver data to base station. Next set of data has to wait on source node until the mobile node returns back or new mobile node is available. This will create high volume of traffic in source node. Increasing number of mobile nodes helps faster delivery of data packets to base station but increases overall cost of sensor network.

CHAPTER IV

METHODOLOGY, MODELS AND EXPERIMENTAL RESULTS

The purpose of this research is to develop a mobility control scheme for power – aware resource management for sensor actuator network. Proposed mobility scheme will increase throughput of the network and decreases average delivery time of data packets to base station. This scheme conserves power and increases lifetime of sensor network and mobile nodes and helps transfer more data to base station.

4.1. Methodology

Single mobile actuator node can be used to transfer data packets to a base station. If a mobile node takes *t* time to deliver a packet to a base station, then next data packet need to wait for time *2t* as mobile node needs to travel back and forth as shown in Fig 4.1.1. Increasing the number of mobile nodes will help decrease delivery time. The proposed mobility pattern of mobile sensor nodes is based on data pipelining scheme.

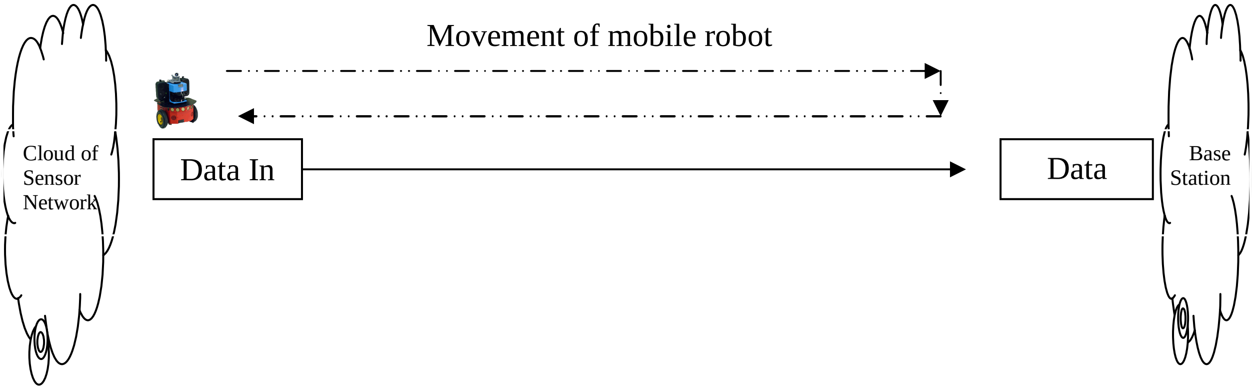


Fig 4.1.1: Data transfer between sink node and base station

Data pipelining is a continuous flow of data in network and one common example is assembly line. Multiple numbers of mobile nodes transfers data packet in pipeline fashion and passes data packet to each other Fig 4.1.2. Mobile nodes are arranged in accordance to total number of mobile node and total distance to base station. A mobile node travels certain distance and meet with adjacent mobile node and passes data packets to the node.

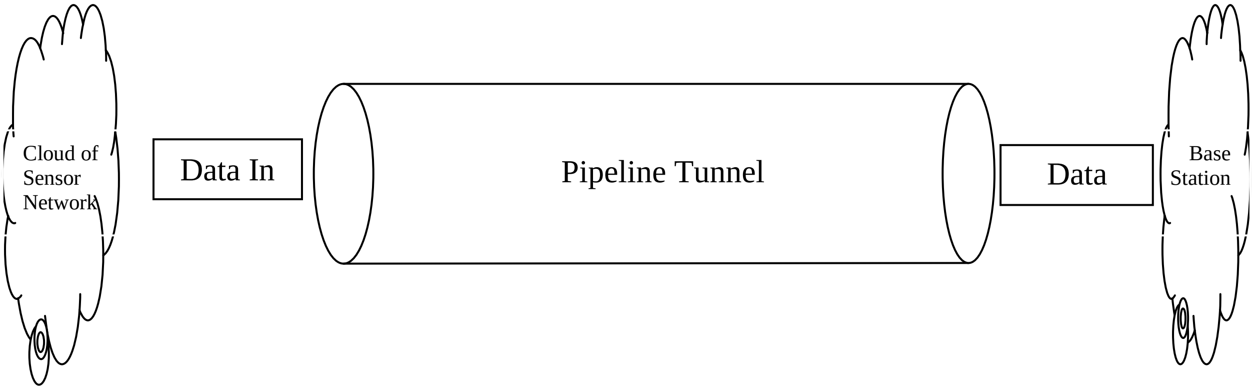


Fig 4.1.2: Data travelling in pipeline tunnel.

In pipelining scheme, first, mobile node collects data from source node, Fig 4.1.3 (a), but rest of mobile node remain in their respective places. The first node then travels certain distance and passes data to second mobile node, Fig 4.1.3 (b).

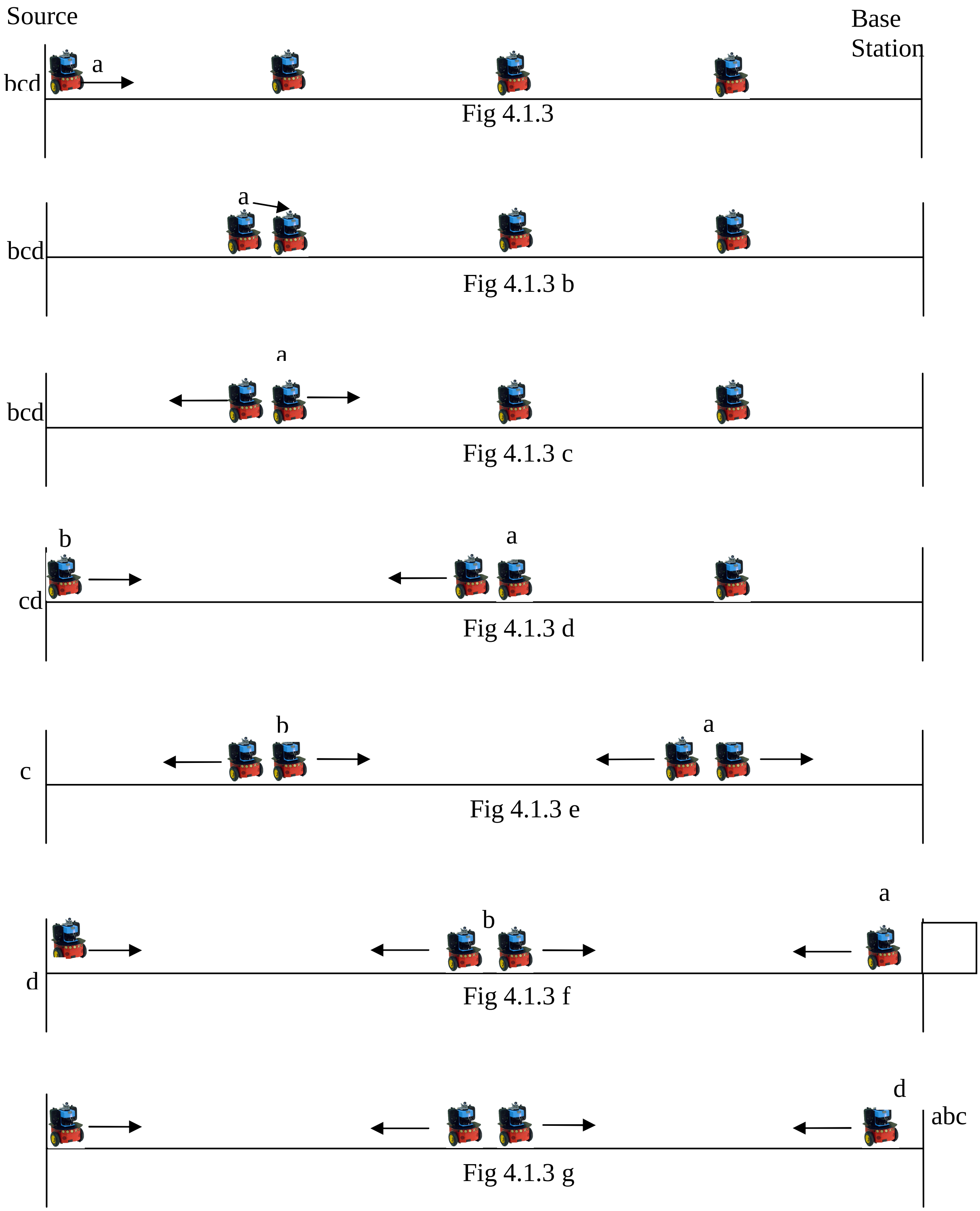


Fig 4.1.3: Design of mobile sensor actors in pipelining environment

The second mobile node then travels certain distance, meets the third mobile node and passes the data to the third mobile node. In the mean time the first mobile node travels back and collects another set of data from the source node as shown in Fig 4.1.3 (d). This back and forth movement of mobile node continues and data keeps on travelling until it reaches the base station as shown in Fig 4.1.3 (f). This creates pattern of movement in mobile nodes, one set of mobile nodes travels forward and other set of mobile nodes travel backward and vice versa in the next step. Fig 4.1.3 (e) shows one set of pattern and Fig 4.1.3 (f) shows another set of pattern in this pipelining environment.

This mobile scheme helps to transfer more data in steady rate to base station in limited time interval. Mobile nodes travel certain distance instead of travelling entire distance which also helps to conserve energy of mobile nodes.

The distance between mobile nodes depends on number of mobile nodes, environment and wireless range of each mobile node. So, each mobile need to travel certain distance *m*,

$$m = \frac{d}{n} - r$$

Where, *m* is the distance mobile nodes need to travel, *d* is the total distance, *n* is the number is available mobile nodes and *r* is the wireless range

4.2 Energy model

Energy cost of mobile actuator node depends on mobility of mobile nodes and communication between mobile nodes. Unit of energy is calculated in SI unit – Joule. If a mass of 1 kg travelling at acceleration rate of 1 m/s^2 and covers 1 meter distance, then energy dissipated by system is 1 Joule.

$$1\ J = 1\ Kg.\ (m/s)^2$$

$$= 1\ Kg.\ m/s^2.m$$

$$= \text{Weight . Acceleration. Distance}$$

where, *m* is distance in meter.

Energy model consists of energy dissipation during motion of mobile nodes and communication cost of mobile nodes. Energy dissipation for motion is

$$E_m = m\ .\ e_{m,}$$

where, *m* is the total distance each mobile need to travel and e_m is the movement parameter and depends on weight, gravity, friction and acceleration of mobile node [18]. Also, the communication cost

$$E_c = m\ .\ e_c$$

where, e_c is the energy consumed by transmitting one bit data over one meter.

Total energy consumed by mobile nodes is

$$E = E_m + E_c.$$

Where, E_m is the energy required for motion and E_c is the energy required for communication.

4.3 Algorithm generating deployment and data pipelining

Deployment and data pipelining of mobile nodes are implemented in algorithm 1. Mobile nodes are deployed depending on total distance of pipeline and number of mobile nodes. Initially, each mobile node is allocated its location in pipelining and energy. Initial distance between mobile nodes is calculated as discussed in energy model 4.3. Total energy cost of mobile nodes depends on energy dissipated both during mobility of mobile nodes and the communication cost of the system as in step 8 – 11. E_m is mobility cost and E_c is communication cost of the network.

```
Algorithm 1: MOBILE-NODE-DEPLOYMENT (d, n)
1.  Initialize total distance (d), number of mobile node(n)
2.  for each i do
3.          Create  mobile node i
4.              Calculate the location for mobile node
5.              Initialize energy for mobile node
6.  End for
7.  Calculate the distance, mobile node need to travel; m ← $\frac{d}{n} - r$ ; where, m is
    distance mobile node need to travel, n is number of mobile node,  r is wireless
    range
8.  Calculate energy dissipation
9.          Energy cost is the mobility cost and communication cost of mobile node;
10.         Mobility cost is total energy require to travel certain distance;
             $E_m \leftarrow e_m \times m$
11.         Communication cost is total energy require to transfer data packet;
            $E_c \leftarrow e_c \times m$
```

Mobile node pipelining algorithm is implemented to forward data packets in a pipelining fashion. Data packets are passed through different available mobile nodes one at a time. Algorithm 2 is divided in two cycles. In the first cycle, source nodes always transfer data packets to the first mobile node as in step 16 – 17. Odd numbers of mobile nodes always collect data from neighboring even mobile nodes (buffer) and even mobile nodes always transfer data packets to buffer. In the second cycle, source nodes always wait for next cycle as the first node travels certain distance and has to transfer data to the second mobile node. Therefore odd mobile nodes always transfer data packets to even mobile nodes. Depending on the total number of available mobile nodes, the last mobile node transfers data packets to base station. For the first cycle, if number of mobile nodes is even, the last mobile transfers data packets to base station and for the second cycle, if number of mobile nodes is odd, the last mobile node transfers data packets to base station.

```
Algorithm 2: MOBILE-NODE-PIPELINING
while flag = 0, then
  for each cycle mobile node moves forward and backward, then
    if Cycle 1, then
    for each mobile, then
      odd node: transferring data from buffer and even node: data to buffer
      buffer ← Sink-node-data   /* transfer data to base station – buffer
      if node is odd, then
        data [i] ← buffer; copying data from buffer
        $E \leftarrow E - E_c - E_m$; Energy cost
      end if
      if node is even, then
        buffer ←data [i]; copying data to buffer
        $E \leftarrow E - E_c - E_m$; Energy cost
      end if
      if total number of node is even, then
        base node ← data [i]
      end if
    end for
    end if
    if Cycle 2, then
    for each mobile node, then
      odd node: transferring data to buffer and even node: data from buffer
      if node is even, then
        data [i] ← buffer; copying data from buffer
        $E \leftarrow E - E_c - E_m$; Energy cost
      end if
      if node is odd, then
        buffer ←data [i]; copying data to buffer
        $E \leftarrow E - E_c - E_m$; Energy cost
      end if
```

```
            if total number of node is odd, then
                base node ← data [i]
            end if
        end for
        end if
        if E < 0, then
            flag ← 0
        end if
    end for
end while
```

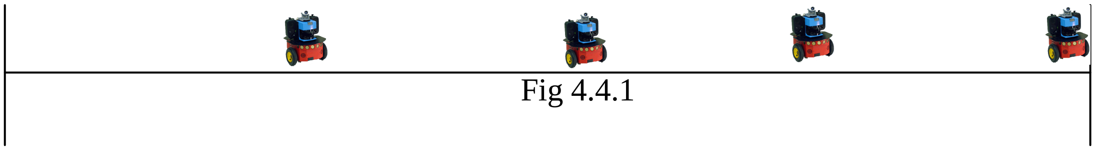


Fig 4.4.1

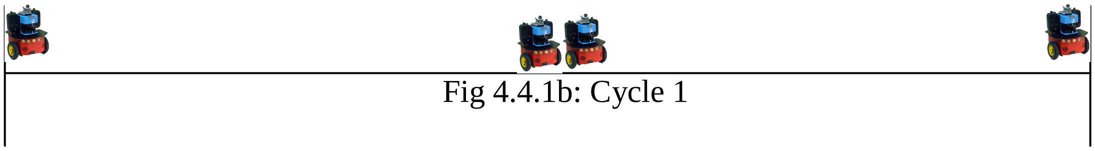


Fig 4.4.1b: Cycle 1

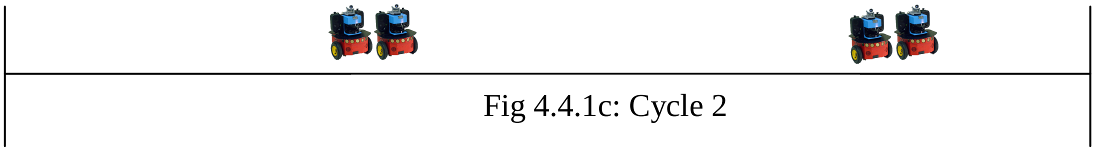


Fig 4.4.1c: Cycle 2

Fig 4.3.1: Pipelining for even number of mobile node

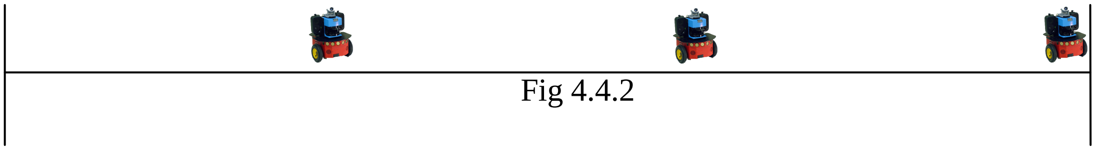


Fig 4.4.2

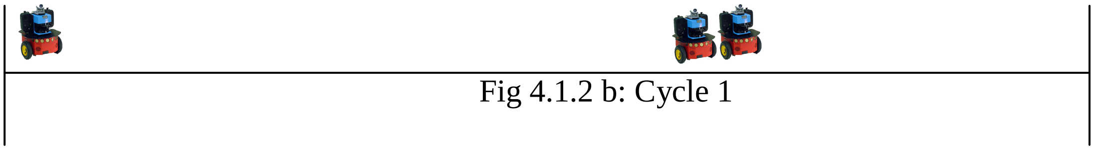


Fig 4.1.2 b: Cycle 1

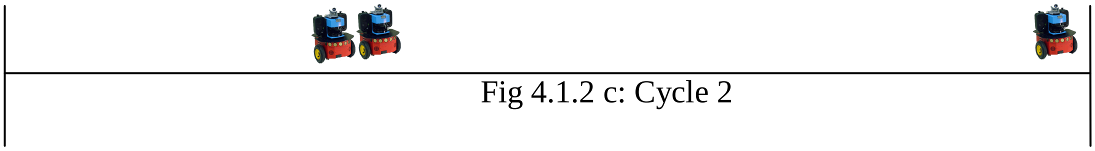
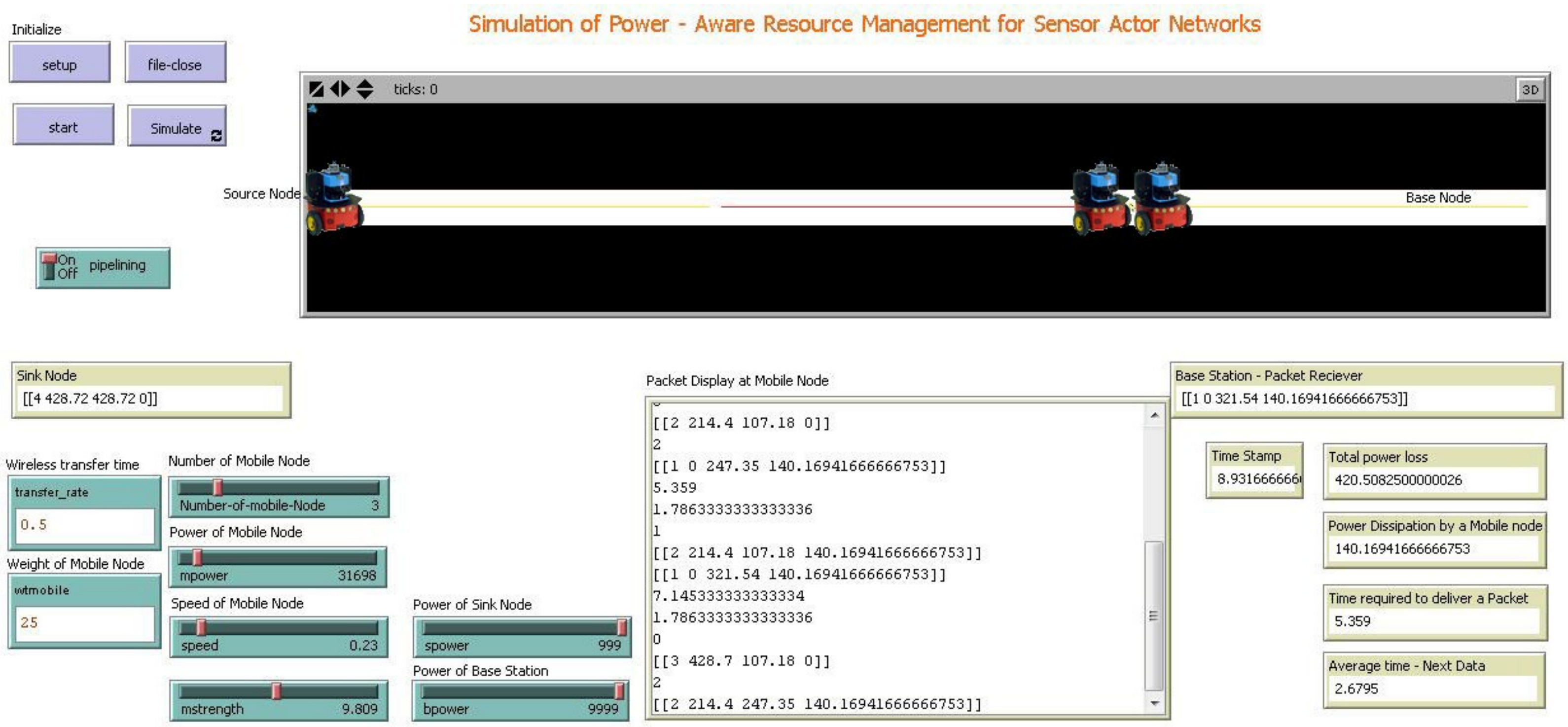


Fig 4.1.2 c: Cycle 2

Fig 4.3.2: Pipelining for odd number of mobile node

4.4 Experimental results

In experimental setup, different number and weight of mobile nodes were varied and tested in these scenarios. Each mobile node holds data packets and travel at certain speed and distance. A simulation platform was developed in Netlogo. Pipelining environment was implemented as discussed in section 4.1.3 and GUI was developed for this simulation as shown in Fig 4.4.1. GUI for source node transferring data packets is shown in Fig 4.4.1a and data packets being transferred between actor nodes are shown in Fig 4.4.1b.



Sink Node: [[4 428.72 428.72 0]]

Base Station - Packet Reciever: [[1 0 321.54 140.16941666666753]]

Wireless transfer time: transfer_rate 0.5

Weight of Mobile Node: wtmobile 25

Number of Mobile Node: Number-of-mobile-Node 3

Power of Mobile Node: mpower 31698

Speed of Mobile Node: speed 0.23

mstrength 9.809

Power of Sink Node: spower 999

Power of Base Station: bpower 9999

Packet Display at Mobile Node

```
[[2 214.4 107.18 0]]
2
[[1 0 247.35 140.16941666666753]]
5.359
1.7863333333333336
1
[[2 214.4 107.18 140.16941666666753]]
[[1 0 321.54 140.16941666666753]]
7.1453333333333334
1.7863333333333336
0
[[3 428.7 107.18 0]]
2
[[2 214.4 247.35 140.16941666666753]]
```

Time Stamp: 8.931666666

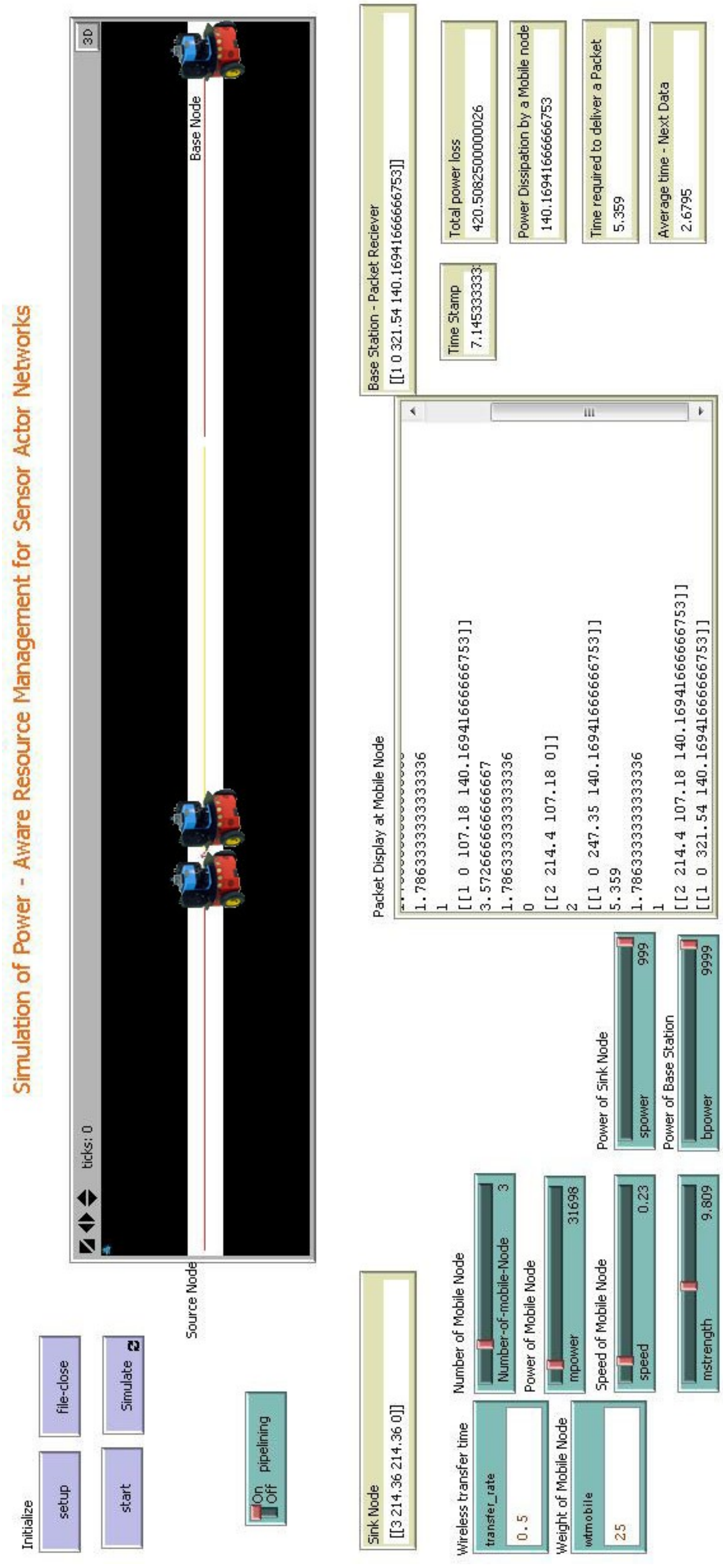
Total power loss: 420.5082500000026

Power Dissipation by a Mobile node: 140.16941666666753

Time required to deliver a Packet: 5.359

Average time - Next Data: 2.6795

Fig 4.4.1a GUI for the simulator



Sink Node: [[3 214.36 214.36 0]]

Wireless transfer time: transfer_rate 0.5

Weight of Mobile Node: wtmobile 25

Number of Mobile Node: Number-of-mobile-Node 3

Power of Mobile Node: mpower 31698

Speed of Mobile Node: speed 0.23

mstrength 9.809

Power of Sink Node: spower 999

Power of Base Station: bpower 9999

Packet Display at Mobile Node

```
1.7863333333333336
1
[[1 0 107.18 140.16941666666753]]
3.5726666666666667
1.7863333333333336
0
[[2 214.4 107.18 0]]
2
[[1 0 247.35 140.16941666666753]]
5.359
1.7863333333333336
1
[[2 214.4 107.18 140.16941666666753]]
[[1 0 321.54 140.16941666666753]]
```

Base Station - Packet Reciever: [[1 0 321.54 140.16941666666753]]

Time Stamp: 7.145333333

Total power loss: 420.50825000000026

Power Dissipation by a Mobile node: 140.16941666666753

Time required to deliver a Packet: 5.359

Average time - Next Data: 2.6795

Fig 4.4.1b GUI for the simulator

Simulation was performed by varying number of mobile nodes for both traditional and pipelining mobility as shown in fig 4.4.2 and fig 4.4.3. Fig 4.4.2 shows traditional mobility pattern and data packets are transferred by individual mobile nodes the entire distance. Fig 4.4.3 shows the proposed mobility pattern and data packets are transferred in pipelining fashion. Mobile nodes moves back and forth and data packets are transferred to neighboring mobile nodes.

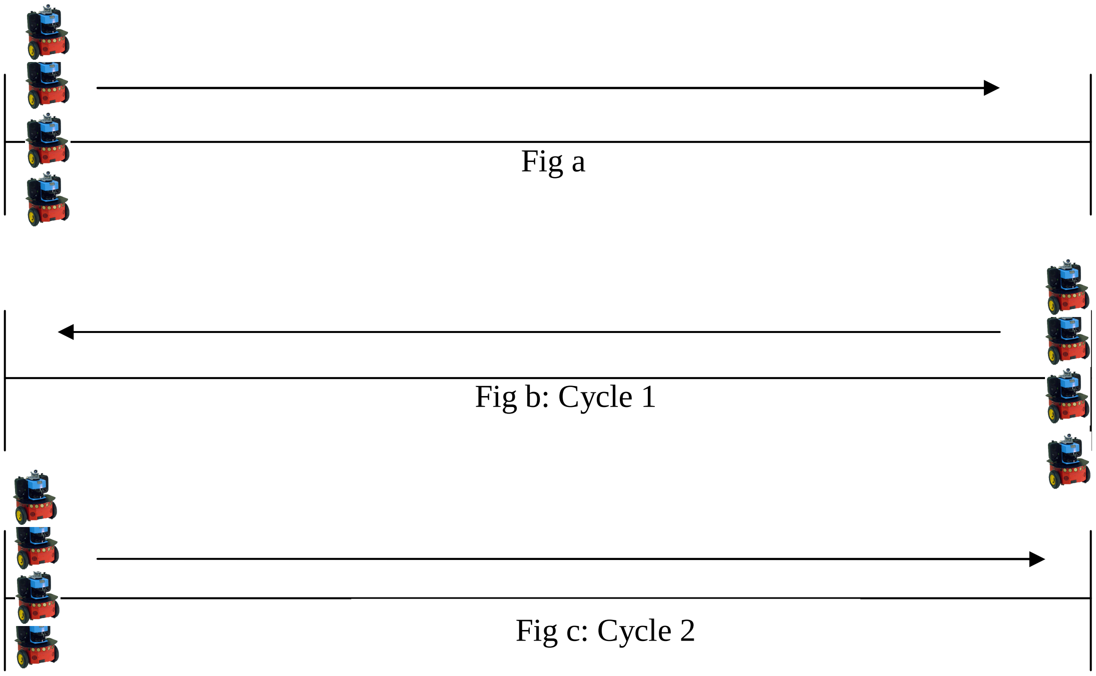


Fig 4.4.2: Traditional mobility pattern of mobile nodes

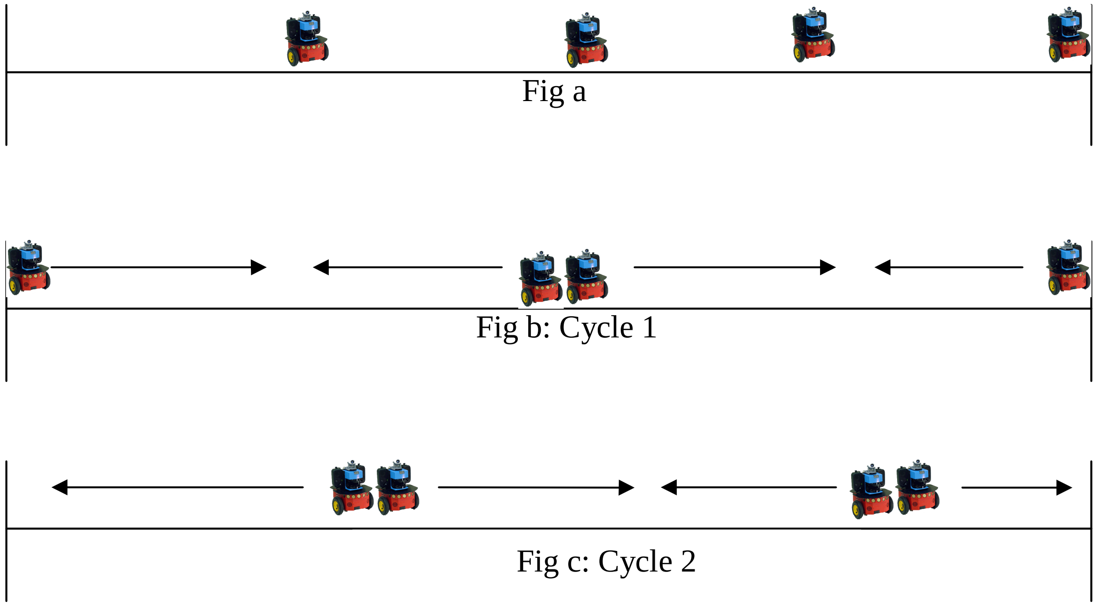


Fig 4.4.3: Pipelining mobility pattern of mobile nodes

Weight of mobile nodes was assumed to be 25 kg, speed 0.23 m/s, wireless communication distance 9.809 m and total distance be 100 meter. Simulations were performed for both traditional mobility pattern and pipelining mobility pattern. Table 4.4.1 shows time required to deliver a packet and obtained by varying number of mobile nodes. Time required to deliver a packet remains constant with traditional mobility pattern as single mobile node travels the entire distance and is responsible to transfer a packet. Increasing the mobile nodes doesn’t affect the delivery time. Proposed mobility pattern with pipelining scheme decreases the total time a packet linearly. As increasing the number of mobile nodes results less travel time and packets are transferred to mobile nodes. This increase in mobile nodes increases the delivery time of the packet to base station as shown in Table 4.4.1 and Fig 4.4.4.

Number of Mobile nodes	Time required to deliver a packet	
	Non pipelining	Pipelining
1	6.61	6.61
2		5.98
3		5.35
4		4.72
5		4.10
6		3.47
7		2.84
8		2.21
9		1.58
10		0.95

Table 4.4.1: Time to deliver packets – varying number of mobile nodes

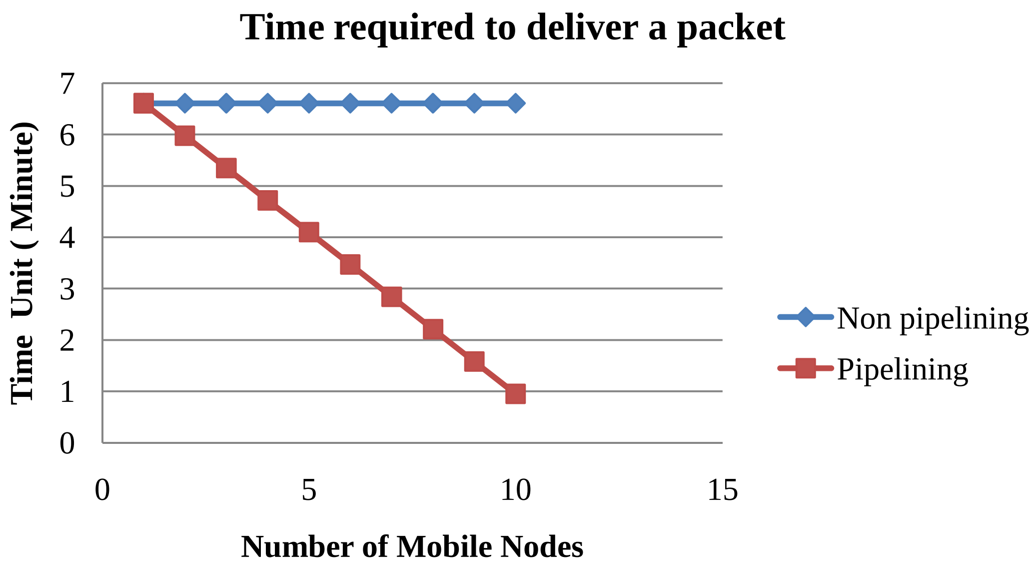


Fig 4.4.4: Time to deliver packets – varying number of mobile nodes

Table 4.4.2 shows the total energy required to deliver a packet by varying number of mobile nodes. In traditional mobility pattern, increasing the mobile nodes does not affect the energy consumption. Mobile nodes need to travel entire distance to deliver a packet. Proposed mobility pattern decreases energy consumption as increasing the number of mobile nodes decreases the travel time and packets are being passed to its neighboring nodes. As number of mobile nodes increases, total distance mobile nodes travel decreases and will increase in wireless transfer distance, this will decrease the energy required to deliver numbers of packet in pipelining strategy as shown in Table 4.4.2 and Fig 4.4.5

Number of Mobile nodes	Energy required to deliver a packet		Energy required to deliver 10 packet		Energy required to deliver 20 packet	
	Non pipelining	Pipelining	Non pipelining	Pipelining	Non pipelining	Pipelining
1	523.50	523.50	9946.55	9946.55	20416.61	20416.61
2		472		9440.11		18880.22
3		420.50		8830.67		17240.84
4		369.01		8118.24		15240.84
5		317.51		7302.81		13653.09
6		266.01		6384.39		11704.73
7		214.51		5362.98		9653.366
8		163.02		4238.57		7499.012
9		111.52		3011.16		3011.168
10		60.02		1680.77		2881.32

Table 4.4.2: Energy require to deliver packets – varying number of mobile nodes

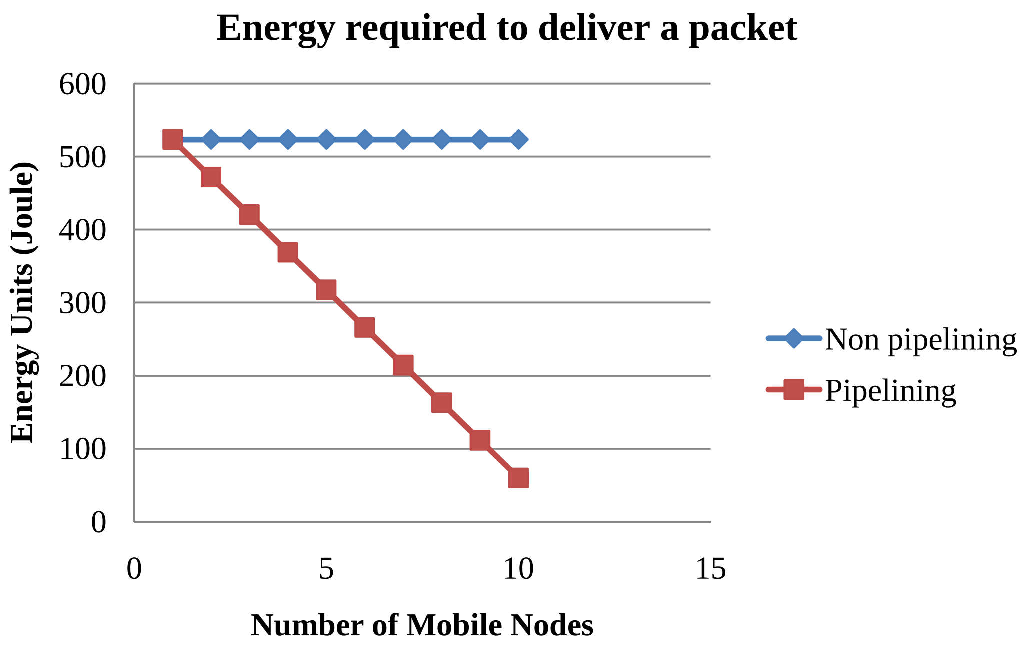


Fig 4.4.5: Energy require to deliver packets – varying number of mobile nodes

Table 4.4.3 shows number of packets delivered in 300 unit time and amount of energy consumed in 300 unit time. Numbers of packets deliver increases with the increase of mobile nodes in pipelining mobility pattern than non pipelining scheme and amount of energy consumption decreases as Table 4.4.3, Fig 4.4.6 and Fig 4.4.7. With increase number of mobile nodes, each mobile nodes travel less distance and result in saving energy and the faster delivery of packet.

Number of Mobile nodes	No. of Packet Delivered (300 Unit time)		Amount of Energy consumed (300 Unit time)	
	Non pipelining	Pipelining	Non pipelining	Pipelining
1	23	23	23557.62	23557.62
2	46	50	47115.25	47200.55
3	69	83	70672.84	70224.88
4	92	125	94230.49	92990.77
5	115	181	117788.1	115892.5
6	138	257	141345.7	137796.5
7	161	366	164903.4	158100.7
8	184	539	188461	171715.8
9	207	848	212018.6	180925.6
10	230	1566	235576.2	188486.4

Table 4.4.3: Number of packets delivered and Energy consumed in 300 unit time

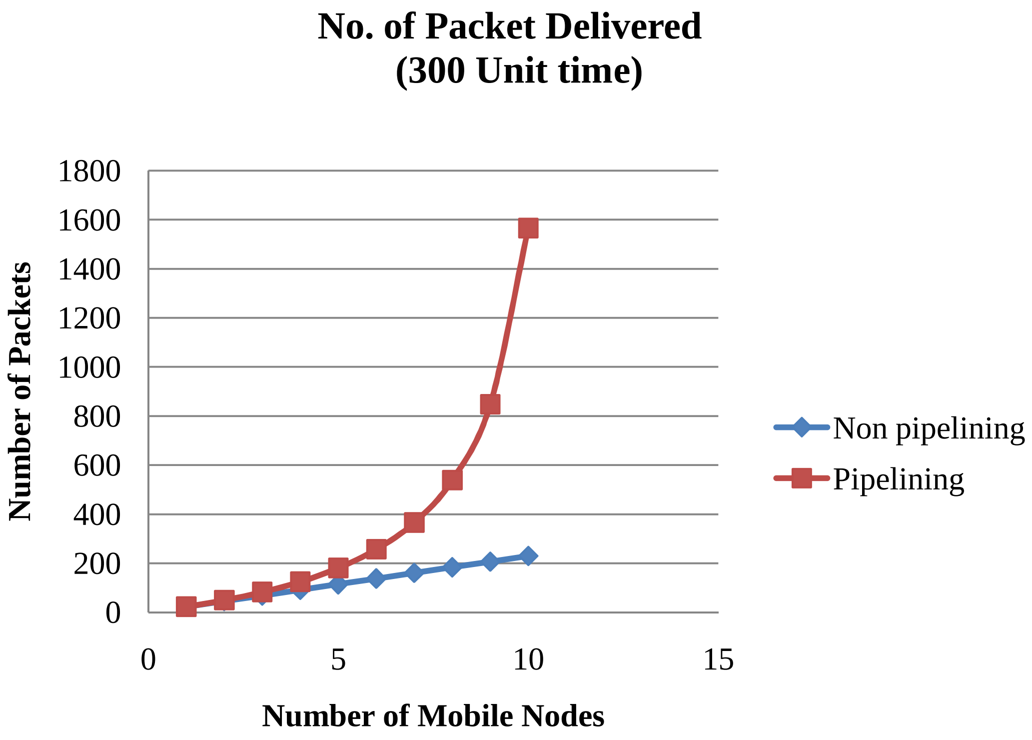


Fig 4.4.6 Number of packets delivered in 300 unit time

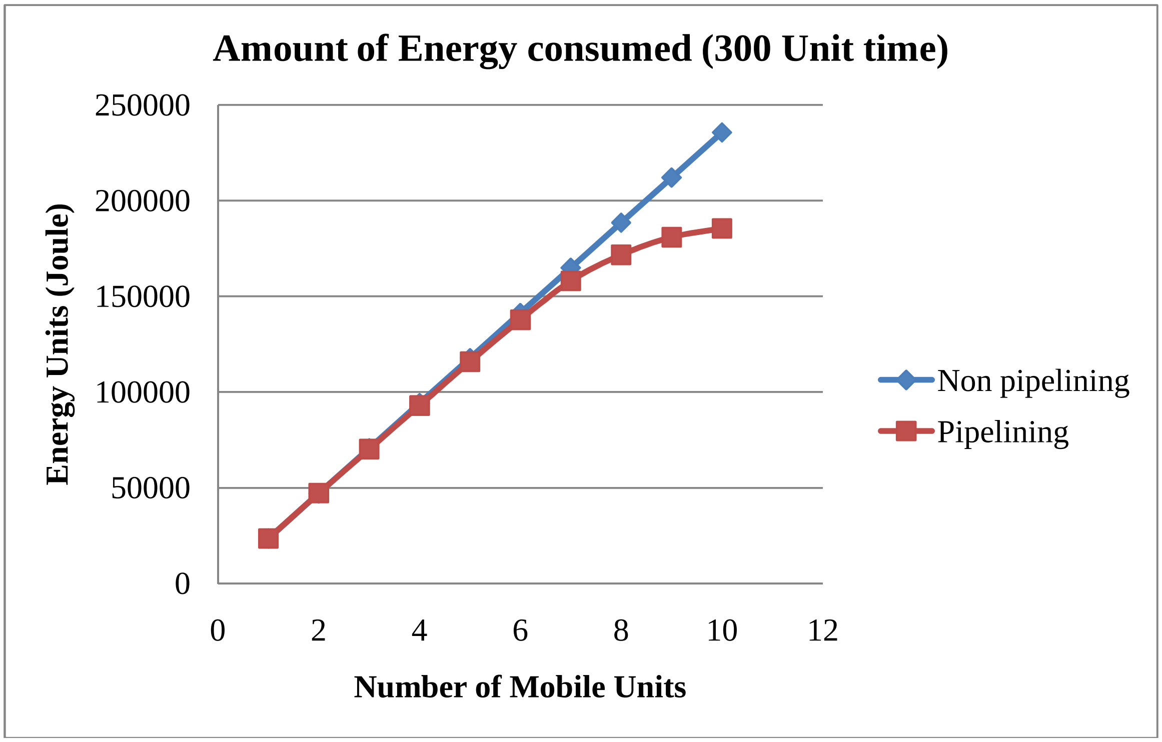


Fig 4.4.7 Energy consumed by mobile nodes in 300 unit time

Table 4.4.4 shows life time of mobile nodes and number of packets delivered. In traditional non pipelining mobility pattern, life time of the mobile node remains constant with the increase of mobile nodes. Proposed pipelining mobility pattern increases the life time of the network as number of mobile nodes is increased as mobile node travels less distance and save energy. Also, number of packet delivered during given energy increase compared to the traditional mobility pattern as shown in Table 4.4.4, Fig 4.4.8 and Fig 4.4.9.

Number of Mobile nodes	Life time with Energy (31698)		Number of Packets delivered with Energy (31698)	
	Non pipelining	Pipelining	Non pipelining	Pipelining
1	397.03	397.03	30	30
2	397.03	401.19	60	67
3	397.03	403.71	90	112
4	397.03	405.59	120	170
5	397.03	409.26	150	248
6	397.03	413.16	180	355
7	397.03	419.80	210	514
8	397.03	430.21	240	774
9	397.03	450.384	270	1276
10	397.03	504.24	300	2636

Table 4.4.4: Life time and number of packets deliver with constant energy of mobile nodes

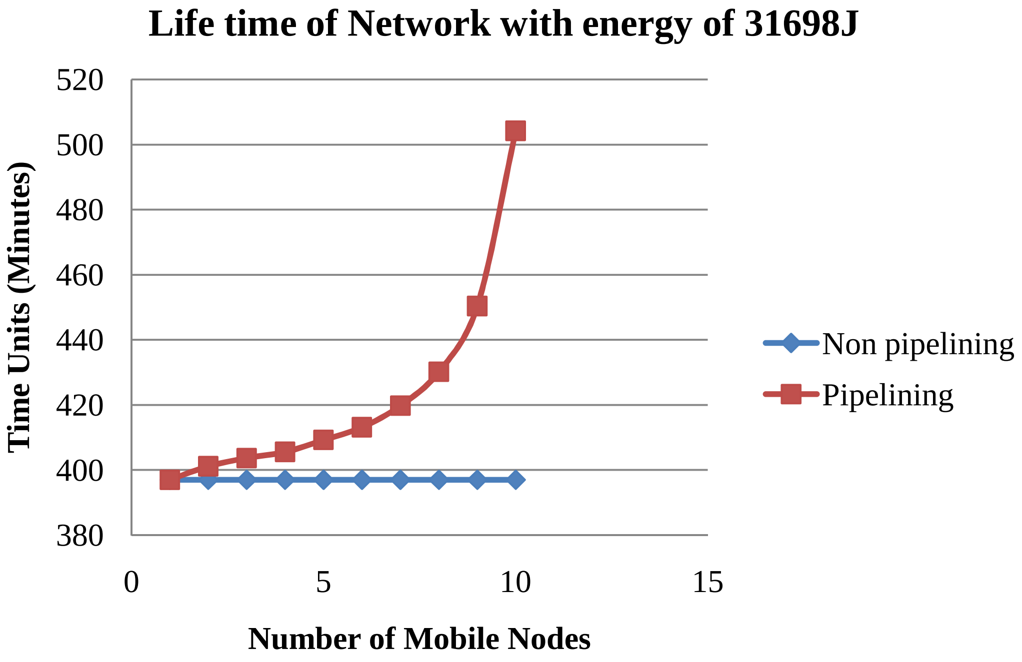


Fig 4.4.8 Life time of mobile nodes with constant energy

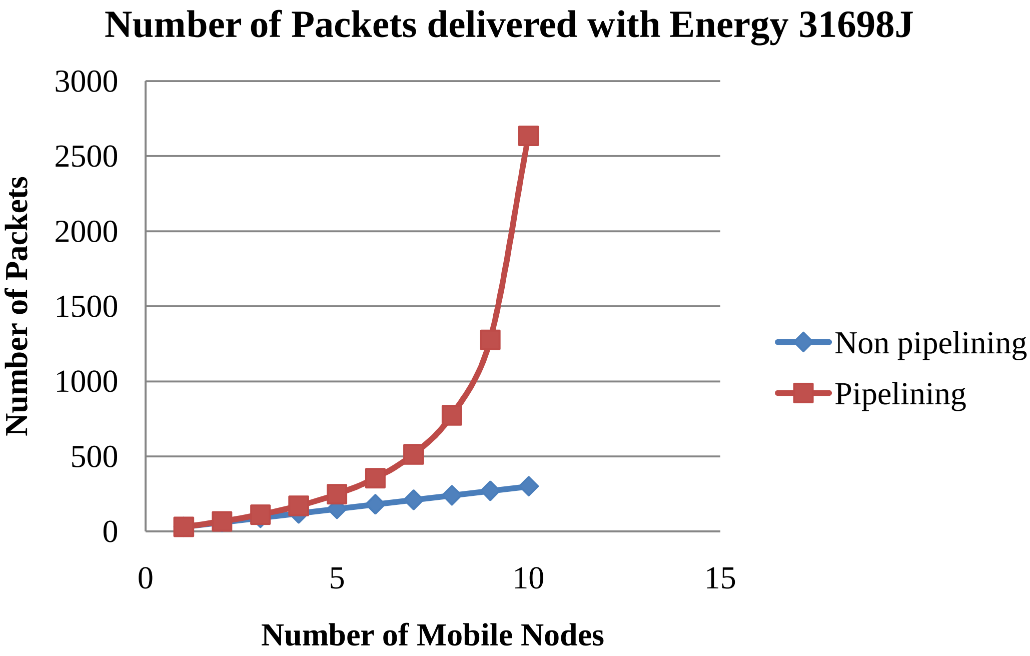


Fig 4.4.9 Number of packets delivers with constant energy for mobile nodes

CHAPTER V

CONCLUSION

4.1 Conclusion

Power management and delivery time of data packet is an important issue in delay tolerant sensor actuator network. Power consumption and delivery time of the system are both important. The deployment strategy used in this research showed that if we increase the number of mobile nodes, average data delivery increases and power consumption decreases.

The strategy is simple and similar to data pipelining. In a pipelining environment, mobile nodes move in pipelining fashion. These mobile nodes collect data from its neighboring mobile nodes; travel a certain distance and transfer data to other mobile nodes. This means mobile nodes do not travel entire distance, but the data packet will still be moving in its path to base station. This strategy will conserve power of mobile nodes and increases the life time of the network. Delivery rate of data packets remains consistent as mobile nodes travel back and forth. Delivery rate of the system depends on how fast mobile nodes transfer data packets to its neighboring mobile nodes. Simulations were designed in NetLogo and performance evaluations have been conducted by varying the number mobile nodes.

The proposed pipelining strategy yields three major benefits. First, as the number of mobile nodes increases, rate of data transfer increases. Second, energy required per data transfer decreases. Finally, life time of the network increases compared to traditional non-pipelining mobility strategy.

4.2 Future Work

Numerous extensions can be carried out in several areas of sensor actuator network. The algorithm can be implemented to incorporate different sensor actuator networks. This will help in communication between different sensor nodes and information can be shared among different networks for real time calculations. In some cases, two different networks might interfere but still can share information. Proposed mobility strategy can be deployed in real scenario of sensor actuator network with mobility of mobile nodes (e.g. robots).Mobility strategy can be tested with different number and speed of mobile nodes. Different type of robots can be tested as they bear different characteristics.

Further, this work can be extended to a larger grid network such as traffic systems and disaster areas. Different schemes of data transfer can be combined together to generate highly efficient sensor actuator network.

REFERENCES

[1] I. Chatzigiannakis, A. Kinalis, and S. Nikoletseas, “Efficient data propagation strategies in wireless sensor networks using a single mobile sink”, Computer Communications, vol. 31, no 5, pp: 896-914, 2008

[2] J. Sushant, F. Kevin, and P. Rabin, “Routing in a delay tolerant network”, Proceedings of 2004 conference on Applications, technologies, architectures, and protocols for computer communications, pp: 145-158, 2004, [online]. Available: http://portal.acm.org/citation.cfm?id=1015484

[3] A. Chakrabarti, A. Sabharwal, and B. Aazhang, “Using predictable observer mobility for power efficient design of sensor networks”, The second International Workshop on Information Processing in Sensor Networks, pp: 129-145, 2003, [online]. Available: http://scholarship.rice.edu/handle/1911/19768

[4] Z. Chen, H. T. Kung, and D. Vlah, “Ad hoc relay wireless networks over moving vehicles on highways”, Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking and computing, Long Beach, CA, USA, ACM, 2001.

[5] R. W. N. Pazzi, and A. Boukerche, “Mobile data collector strategy for delay-sensitive applications over wireless sensor networks”, Computer Communications, vol 31, no 5 pp: 1028-1039, 2008.

[6] C. Liu, and J. Wu, "Routing in a cyclic mobispace", Proceedings of the 9th ACM international symposium on Mobile ad hoc network and computing, pp: 351 – 300, 2008, [online]. Available: http://portal.acm.org/citation.cfm?id=1374665

[7] J. A. Davis, A. H. Fagg, and B. N. Levine, "Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks", Proceedings of the 5th international symposium on Wearable Computers, pp: 141, 2001.

[8] M. Tariq, M. Ammar, and E. Zegura, "Message ferry route design for sparse ad hoc networks with mobile nodes", Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing, pp: 37 – 48, 2006.

[9] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks", Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing, 2004.

[10] T. Wart, C. Crossman, "The design and evaluation of a mobile sensor/actuator network for autonomous animal control", Proceedings of the 6th international conference on Information processing in sensor networks, pp: 206 – 215, 2007, [online]. Available: http://portal.acm.org/citation.cfm?id=1236389&coll=GUIDE&dl=GUIDE&ret=17

[11] A. Venkateswaran, V. sarangen, T. Porta, and R. Acharya, "A mobility prediction based relay deployment framework for conserving power in manets", Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems, pp: 209 – 216, 2007.

[12] A. Camilli, C. E. Cugnasca, A. M. Saraiva, A. R. Hirakawa, and P. Correa, "From wireless sensors to field mapping: Anatomy of an application for precision agriculture", Computers and Electronics in Agriculture, vol 58, no 1, pp: 25-36, 2007

[13] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey”, Computer Networks, vol. 38, no 4, pp: 393-422, 2002.

[14] I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis, “Smart dust protocols for local detection and propagation”, Proceedings of the second ACM international workshop on Principles of mobile computing, pp: 9 -16, 2002.

[15] I. F. Akyildiz, and I. H. Kasimoglu, “Wireless sensor and actor networks: research challenges”, Ad Hoc Networks, vol 2, no 4, pp: 351-367, 2004.

[16] T. Melodia, D. Pompili, V. C. Gungor, and I. F. Akyildiz, “A distributed coordination framework for wireless sensor and actor networks”, Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing, pp: 99 – 110, 2005.

[17] S. N. Pakzad, G. L. Fenves, S. Kim, and D. E. Culler, “Design and Implementation of Scalable Wireless Sensor Network for Structural Monitoring”, Journal of Infrastructure Systems, vol 14, no 1, pp: 89-101, 2008.

[18] C. C. Ooi, and C. Schindelhauer, “Detours Save Energy in Mobile Wireless Networks”, 2008, [online]. Available:
http://www.springerlink.com/content/p24382355171q6n2/

[19] S. Medhekar, R. Howard, W. Trappe, and Y. Zhang, “Mining Joules and Bits: Towards a Long Life Pervasive System”, IEEE International Symposium on Parallel and Distributed Processing, pp: 1-8, 2008.

[20] Y. Mei, YH. Lu, Y. C. Hu, and C.S.G. Lee, “Deployment of Mobile Robots With Energy and Timing Constraints”, IEEE Transactions on Robotics, vol 22, no 3, pp: 507-522, 2006.

[21] Z. Khalid, G. Ahmed, and N. M. Khan, “Impact of Mobile Sink Speed on the Performance of Wireless Sensor Networks”, Journal of Information and Communication Technology, vol 1, no 2, pp: 49-55, 2007, [online]. Available: http://www.biztek.edu.pk/downloads/JICT/6%20Impact%20of%20Mobile%20Sink1.pdf

[22] T. Bokareva, N. Bulusu, and S. Jha, “SASHA: Toward a Self-Healing Hybrid Sensor Network Architecture”, The Second IEEE Workshop on Embedded Networked Sensors, pp: 71-78, 2005.

[23] R. C. Shah, and J. M. Rabaey, “Energy Aware Routing for Low Energy Ad Hoc Sensor Networks”, Wireless communications and Networking Conference, vol 1, pp: 350-355, 2002.

[24] S. Ni, Y. Tseng, Y. Chen, and J Sheu, “The Broadcast Strom Problem in Mobile Ad Hoc Network”, Proceddings of the 5th annual ACM/ IEEE International Conference on Mobile Computing and Networking, pp: 151-161, 1999.

[25] F. Schmidt, G. Scholl, A. Anders, H. Korber, and H. Wattar, “RF-Embedding of Energy-Autonomous Sensor Networks”, Germany, 2006, [online]. Available: http://stinet.dtic.mil/cgi-bin/GetTRDoc?AD=ADA479950&Location=U2&doc=GetTRDoc.pdf

[26] N. Banerjee, M. D. Corner, D. Towsley, and B. N. Levine, “Relays, Base Stations, and Meshes: Enhancing Mobile Networks with Infrastructure”, Proceedings of 14th AVM international Conference on Mobile Computing and Networking, pp: 81-91, 2008, [online]. Available: http://portal.acm.org/citation.cfm?id=1409955

[27] J. Leguay, T. Friedman, and V. Conan, “DTN Routing in a Mobility Pattern Space”, Proceedings of 2005 ACM SIGOMM workshop on Delay-tolerant Networking, pp: 276-283, 2005, [online]. Available: http://portal.acm.org/citation.cfm?id=1080139.1080146

[28] N. Xu, S. Rangwala, K.k. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, “A Wireless Sensor Network for Structural Monitoring”, Proceedings of the 2nd International conference on Embedded Networked Sensor Systems, pp: 13-24, 2004

[29] J. Agre, and L. Clare, “An integrated Architecture for Cooperative Sensing Networks”, Computer, vol 33, no 5, pp:106-108, 2002, [online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=841788

[30] A. Durresi, V. Paruchuri, and L. Barolli, “Delay-Energy Aware Routing Protocol for Sensor and Actor Networks”, Proceedings of 11th International Conference on Parallel and Distributed Systems, vol 1, pp: 20-22, 2005, [online]. Available:
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1531141

[31] A. Meissner, T. Luckenbach, T. Risse, T. Kirste, and H. Kirchner, “ Design Challenges for an Integrated Disaster Management Communication and Information Systems”, The First IEEE Workshop on Disaster Recovery Networks, vol 24, 2002.

APPENDICES

Simulation environment was designed in NetLogo 4.0.4. Implementation of algorithm 2 – mobile node pipelining is included in function phase 1 and phase 2. Mobile node transfers data from source to base station. In phase 1, source node always transfers data to first mobile node and base station receives data depending on number of available mobile node. Even number of mobile node always transfers data to odd number of mobile node. In phase 2, odd number of mobile node transfers data to even number of mobile node an travels certain distance depending on the position of mobile node

```
// Function for phase 1 mobility

to phase1
  set i 0
  ask snode number-of-mobile-node [              ;; Read the value from source node
    if(slist != [])
      [
          set mbuffer last slist
           ;; generate the data for sink node
          generate-data

          set tbuffer last mbuffer
          set tbuffer t_out
          set mbuffer replace-item 1 mbuffer (precision tbuffer 1)
          set mbuffer replace-item 2 mbuffer (precision tbuffer 2)
```

```
           ;;set t t + (precision (t_travel )2)
           set slist but-last slist                              ;; removing the data from source node
           set spwr spwr - (edata )                                  ;; pwr change in source node
           show "t"
           show t
           show t_travel
       ]
     ]


   while [i <= nm] [
      ask mnode i[
        set m_travel (dmnode - (2 * mstr))                    ;; distance mobile node need to travel
        set t_travel (precision (m_travel / mspeed ) 2)    ;; time taken by mobile node to travel

        ifelse ((i mod 2) != 0)                        ;; i = 1 (second)
        [

          set mbuffer last mlist                                         ;; copying data to the buffer
          if((first mbuffer) != "x")
          [
            set mpwrprev mpwr
            set mpwr mpwr - (m_travel * etrav) -  (edata * mstr)           ;; decrease of power
            show "power"
            show m_travel * etrav
            show etrav
            show m_travel
         ]

        ]
        [
          if((first mbuffer) != "x")
          [
            set tbuffer last mbuffer
            set t_time transfer_rate * mstr
            set tbuffer tbuffer + t
            set mpwrprev mpwr
            set mpwr mpwr -  (edata * mstr)
            set mpwrprev mpwr
             set mpwr mpwr - (m_travel * etrav)
             if( i != 0)
             [
            set mbuffer replace-item 2 mbuffer (precision tbuffer 2)
             ]
```

```
          set mlist fput mbuffer mlist                    ;; copying data to the mobile node
          set mlist but-last mlist
           if((first mbuffer)!= "x")
            [
              output-print i
              output-print mlist
              output-print mpwr                           ;; removing the data from sink node
            ]
        ]
      ]
      set i i + 1
    ]
  ]


   ;; if number of mobile node is even - copy the last buffer to base station
  if((nm + 1) mod 2  = 0)
  [
    ask bnode (bn - 1)[
    if((first mbuffer) != "x")
    [                                         ;; bn is position of node - 1
      set tbuffer last mbuffer
      set tbuffer tbuffer + t
      set mbuffer replace-item 2 mbuffer (precision t_out 2)

      set blist fput mbuffer blist                    ;; copying the buffer value to base station node
      set bpwr bpwr -  (edata )
      set bwrite 1
    ]
    ]
  ]

  set icount icount + 1
  set clk2 clk2 + t
  display-out
end

// Function for phase 2 mobility

to phase2
  set i 0
  while [i <= nm] [
    ask mnode i[
      set m_travel (dmnode - (2 * mstr))
```

```
     set t_travel (precision (m_travel / mspeed ) 2)
     ifelse ((i mod 2) = 0)
     [
       set mbuffer last mlist
       if((first mbuffer) != "x")
       [

         set mpwrprev mpwr
         set mpwr mpwr - (m_travel * etrav)-  (edata * mstr)
       ]

     ]
     [

      ifelse((first mbuffer) != "x")
       [
         set t_time transfer_rate * mstr
         set tbuffer last mbuffer
         set tbuffer tbuffer + t

         set mbuffer replace-item 2 mbuffer (precision tbuffer 2)
         set mpwrprev mpwr
         set mpwr mpwr -  (edata * mstr)
         set mpwr mpwr - (m_travel * etrav)
       ]

       [
           set mpwr mpwr - (m_travel * etrav)
       ]

       set mlist fput mbuffer mlist                              ;; copying data to the buffer
       set mlist but-last mlist
       if((first mbuffer)!= "x")
       [
         output-print i
         output-print mlist
         output-print mpwr                              ;; removing the data from sink node
       ]
     ]
      set i i + 1
     ]
 ]

;; if number of mobile node is even - copy the last buffer to base station

  if((nm + 1) mod 2 != 0)
```

```
 [
   ask bnode (bn - 1)[
      if((first mbuffer) != "x")
     [                                              ;; bn is position of node - 1
     set t_time transfer_rate * mstrength
     set tbuffer last mbuffer
     set tbuffer tbuffer + t
     set mbuffer replace-item 2 mbuffer (precision t_out 2)

     set blist fput mbuffer blist                   ;; copying the buffer value to base station node
     set bpwr bpwr -  (edata )
     set bwrite 1

     ]
   ]
 ]


  set icount icount + 1

  display-out
  set clk2 t_out                                ;; screen shot of data on output monitor


end

to data_calc
  set mspeed speed
  set etrav wtmobile * (mspeed)             ;; energy required for traveling distance in Joule
  set edata  transfer_rate                  ;; energy required for data transfer
end



to display-out                                 ;; screen shot of data on output monitor
 set i 0
 set avgmpwr 0
 set mpwr_buffer 0
 ]
 set mpwr_buffer 0
 while [i <= nm][                                ;; displaying node value
   ask mnode i[



     file-type mpwr
```

```
    file-type nm

    set mpwr_buffer mpwrprev - mpwr
    set avgmpwr avgmpwr + mpwr_buffer

    set i i + 1
  ]
 file-type "\t"

]
set avgmpwr avgmpwr / (nm + 1)
file-type (t_out / 60)
file-type "\t"
if(bwrite = 1)
[

ask bnode (bn - 1)[
output-print blist
if(blist != [])
[
  set graphtemp blist
  set graphtemp first graphtemp
  set graphno first graphtemp
  file-type graphno
  file-type "\t"

  set graphtemp but-first graphtemp

  set graphin first graphtemp
  file-type (graphin / 60)

  file-type "\t"
  set graphout last graphtemp

  file-type (graphout / 60)
  file-type "\t"
  if( graphno > ntest)
 [
  plot graphout / 60
  plot graphin / 60
  set avgtout (((avgtout + (graphout - graphin)) / 2) / 60)
  set tint (graphout - graphout_temp)
  set avgtint (((avgtint + tint) / 2) / 60 )
  set bwrite 0
  set ntest graphno
  set graphout_temp graphout
```

```
  ]
  ]
  ]

  ]
  file-type "\n"
end


to generate-data                                    ;; generating data for sink node
  ask mnode 0 [
     set m_travel (dmnode - (2 * mstr))
     set t_travel (precision (m_travel / mspeed ) 2)
          ]

     set t_time (precision (transfer_rate * mstrength) 2)
     set t (precision ( t_time + t_travel + random (t_travel / 10)) 2)




  ask snode number-of-mobile-node[
     set xx first slist                                  ;; copying first list in buffer
     set xxx first xx                                    ;; copying first data of xx list in buffer
     set xxx xxx + 1                                     ;; increasing copied value by 1
     set xx but-first xx                                 ;; removing the first data of list
     set xx fput xxx  xx                                 ;; copying the xxx in the first array of list
     set xx replace-item 1 xx (precision (clk2 + t) 2)
     set xx replace-item 2 xx (precision (clk2 + t) 2)
     set slist fput xx slist                           ;; copying the list in slist
  ]
  end
```

VITA

Abhishek Ghale

Candidate for the Degree of Computer Science

Master of Science

Thesis: POWER – AWARE RESOURCE MANAGEMENT FOR SENSOR – ACTOR NETWORKS

Major Field: Mobile Sensor Actuator Network

Biographical:

Personal Data: Born in Nepal on August 8th, 1981

Education:

Completed the requirements for the Master of Science in Computer Science at Oklahoma State University, Stillwater, Oklahoma in December, 2009.

Received B.E. Degree of Computer Engineering from Pokhara University, Pokhara, Nepal, 2004.

Experience:

Graduate Assistant in Department of CEAT Distance Education, Oklahoma State University, 2007 to 2009

Name: Abhishek Ghale Date of Degree: December, 2009

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: POWER – AWARE RESOURCE MANAGEMENT FOR SENSOR – ACTOR NETWORKS

Pages in Study: 49 Candidate for the Degree of Master of Science

Major Field: Computer Science

Scope and Method of Study: Mobile sensor actuator networks have been applied in a wide range of real-world applications. Because nodes in sensor networks are powered by batteries, two major problems arise. First, the nodes do not last long because of limited power supply. Second, only limited amount of data can be transferred if proper mobility strategies are not designed. In this research, we propose the concept of pipelining of mobile nodes to asynchronously transfer and relay data in the network. Through a number of simulations we examine the design tradeoff of performance and energy cost. Performance evaluation through simulations has been conducted by number of mobile nodes.

Findings and Conclusion: The proposed pipelining strategy yields three major benefits. First, as the number of mobile nodes increases, rate of data transfer increases. Second, energy required per data transfer decreases. Finally, life time of the network increases compared to traditional non-pipelining mobility strategy.

ADVISER'S APPROVAL: Dr. Xiaolin Li