Medium Access Control Protocols And Routing Algorithms For Wireless Sensor Networks

2007

Anirban Bag
University of Central Florida

Find similar works at: https://stars.library.ucf.edu/etd

University of Central Florida Libraries http://library.ucf.edu

Part of the Computer Sciences Commons, and the Engineering Commons

STARS Citation

https://stars.library.ucf.edu/etd/3069

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of STARS. For more information, please contact lee.dotson@ucf.edu.
MEDIUM ACCESS CONTROL PROTOCOLS AND ROUTING ALGORITHMS FOR WIRELESS SENSOR NETWORKS

by

ANIRBAN BAG
M.S. University of Central Florida, 2007

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the School of Electrical Engineering and Computer Science in the College of Engineering and Computer Science at the University of Central Florida Orlando, Florida

Fall Term
2007

Major Professor: Mostafa A. Bassiouni
ABSTRACT

In recent years, the development of a large variety of mobile computing devices has led to wide scale deployment and use of wireless ad hoc and sensor networks. Wireless Sensor Networks consist of battery powered, tiny and cheap “motes”, having sensing and wireless communication capabilities. Although wireless motes have limited battery power, communication and computation capabilities, the range of their application is vast.

In the first part of the dissertation, we have addressed the specific application of Biomedical Sensor Networks. To solve the problem of data routing in these networks, we have proposed the Adaptive Least Temperature Routing (ALTR) algorithm that reduces the average temperature rise of the nodes in the in-vivo network while routing data efficiently. For delay sensitive biomedical applications, we proposed the Hotspot Preventing Routing (HPR) algorithm which avoids the formation of hotspots (regions having very high temperature) in the network. HPR forwards the packets using the shortest path, bypassing the regions of high temperature and thus significantly reduces the average packet delivery delay, making it suitable for real-time applications of in-vivo networks. We also proposed another routing algorithm suitable for being used in a network of id-less biomedical sensor nodes, namely Routing Algorithm for networks of homogeneous and Id-less biomedical sensor Nodes (RAIN). Finally we developed Biocomm, a cross-layer MAC and Routing protocol co-design for Biomedical Sensor Networks, which optimizes the overall performance of an in-vivo network through cross-layer interactions. We performed extensive simulations to show that the proposed Biocomm protocol performs much better than the other existing MAC and Routing
protocols in terms of preventing the formation of hotspots, reducing energy consumption of nodes and preventing network congestion when used in an in-vivo network.

In the second part of the dissertation, we have addressed the problems of habitat-monitoring sensor networks, broadcast algorithms for sensor networks and the congestion problem in sensor networks as well as one non-sensor network application, namely, on-chip communication networks. Specifically, we have proposed a variation of HPR algorithm, called Hotspot Preventing Adaptive Routing (HPAR) algorithm, for efficient data routing in Networks On-Chip catering to their specific hotspot prevention issues. A protocol similar to ALTR has been shown to perform well in a sensor network deployed for habitat monitoring. We developed a reliable, low overhead broadcast algorithm for sensor networks namely Topology Adaptive Gossip (TAG) algorithm. To reduce the congestion problem in Wireless Sensor Networks, we proposed a tunable cross-layer Congestion Reducing Medium Access Control (CRMAC) protocol that utilizes buffer status information from the Network layer to give prioritized medium access to congested nodes in the MAC layer and thus preventing congestion and packet drops. CRMAC can also be easily tuned to satisfy different application-specific performance requirements. With the help of extensive simulation results we have shown how CRMAC can be adapted to perform well in different applications of Sensor Network like Emergency Situation that requires a high network throughput and low packet delivery latency or Long-term Monitoring application requiring energy conservation.
To my parents, Asok Kumar Bag and Debasree Bag
# TABLE OF CONTENTS

LIST OF FIGURES ........................................................................................................... V

CHAPTER 1 INTRODUCTION ........................................................................................1

1.1. Wireless Ad-hoc Networks ...................................................................................... 1
1.2. Wireless Sensor Networks ....................................................................................... 2
   1.2.1. Constraints of Wireless Sensor Networks ......................................................... 3
   1.2.2. Biomedical Sensor Networks ............................................................................ 4
   1.2.3. Habitat-monitoring Sensor Networks ............................................................... 6
   1.2.4. Networks On-Chip ............................................................................................ 7
1.3. Routing Algorithms ................................................................................................. 8
1.4. Medium Access Control Protocols ........................................................................ 11
1.5. Cross-Layer Protocol Design ................................................................................. 12

CHAPTER 2 ENERGY EFFICIENT THERMAL AWARE ROUTING ALGORITHMS
FOR EMBEDDED BIOMEDICAL SENSOR NETWORKS............................................. 14

2.1. Introduction............................................................................................................ 14
2.2. Proposed Algorithms ............................................................................................. 16
   2.2.1. Least Temperature Routing Protocol .............................................................. 16
   2.2.2. Adaptive Least Temperature Routing protocol .............................................. 17
2.3. Simulation.............................................................................................................. 18
   2.3.1. Network of 12 randomly connected nodes ..................................................... 19
   2.3.2. 4x4 Mesh topology ......................................................................................... 23
   2.3.3. 4x4 Mesh Torus network ................................................................................ 26
   2.3.4. 8-node Ring topology ..................................................................................... 30
   2.3.5. Dense network of 50 randomly connected nodes ........................................... 33
   2.3.6. Lifetime of Network ....................................................................................... 36
2.4. Conclusion ............................................................................................................. 38

CHAPTER 3 HOTSPOT PREVENTING ROUTING ALGORITHM FOR DELAY-
SENSITIVE APPLICATIONS OF IN-VIVO BIOMEDICAL SENSOR NETWORKS. 39

3.1. Introduction............................................................................................................ 39
3.2. Proposed Algorithm............................................................................................... 40
3.3. Pseudo Code .......................................................................................................... 43
3.4. Simulation.............................................................................................................. 44
   3.4.1. 4x8 Mesh-torus topology ................................................................................ 46
   3.4.2. Topology consisting of a 3D layout of 27 nodes .............................................. 52
   3.4.3. Topology consisting of 100 densely connected nodes .................................... 55
   3.4.4. Lifetime of a network ..................................................................................... 58
3.5. Conclusion ............................................................................................................. 61

CHAPTER 4 ROUTING ALGORITHM FOR NETWORK OF HOMOGENEOUS AND
ID-LESS BIOMEDICAL SENSOR NODES (RAIN) ..................................................... 63

4.1. Introduction............................................................................................................ 63
4.2. Proposed Algorithm............................................................................................... 65
4.3. Simulation Results ........................................................................................................ 70
   4.3.1. 11x11 regular mesh topology .............................................................................. 72
   4.3.2. Topology consisting of 100 randomly connected nodes ........................................... 82
   4.3.3. Operational life of the network .......................................................................... 88
4.4. Conclusion ..................................................................................................................... 94

CHAPTER 5 BIOCOMM – CROSS-LAYER MAC AND ROUTING PROTOCOL CO-
DESIGN FOR BIOMEDICAL SENSOR NETWORKS. ..................................................... 95
5.1. Introduction ................................................................................................................... 95
5.2. Proposed Architecture ............................................................................................... 96
5.3. Proposed Protocol ..................................................................................................... 99
   5.3.1. Medium Access Control (MAC) ...................................................................... 99
   5.3.2. Routing .......................................................................................................... 101
   5.3.3. Modifications for delay sensitive applications (Biocomm-D) ....................... 102
5.4. Simulation .................................................................................................................. 103
5.5. Conclusion ................................................................................................................ 119

CHAPTER 6 ECO-FRIENDLY THERMAL AWARE ROUTING PROTOCOL FOR
HABITAT MONITORING .............................................................................................. 120
6.1. Introduction ................................................................................................................ 120
6.2. System Topology ...................................................................................................... 121
6.3. Proposed Protocol ................................................................................................... 123
6.4. Simulation ................................................................................................................ 125
6.5. Conclusion ................................................................................................................ 132

CHAPTER 7 TOPOLOGY ADAPTIVE GOSSIP ALGORITHM FOR WIRELESS
SENSOR NETWORKS ..................................................................................................... 133
7.1. Introduction ................................................................................................................ 133
7.2. Proposed Algorithm ................................................................................................. 134
7.3. Simulation ................................................................................................................ 136
7.4. Conclusions ............................................................................................................. 142

CHAPTER 8 A TUNABLE CROSS-LAYER CONGESTION REDUCING MEDIUM
ACCESS CONTROL (CRMAC) PROTOCOL FOR WIRELESS NETWORKS ....... 143
8.1. Introduction ................................................................................................................ 143
8.2. Proposed protocol .................................................................................................... 146
8.3. Simulation Results ................................................................................................. 150
   8.3.1. Scenario A ..................................................................................................... 152
   8.3.2. Scenario B ..................................................................................................... 158
8.4. Conclusion ................................................................................................................ 166

CHAPTER 9 HOTSPOT PREVENTING ADAPTIVE ROUTING ALGORITHM FOR
NETWORKS-ON-CHIP ................................................................................................. 168
9.1. Introduction ................................................................................................................ 168
9.2. Proposed Algorithm ............................................................................................... 170
9.3. Simulation ................................................................................................................ 173
   9.3.1. Two dimensional mesh topology of 4x8 routers ............................................. 174
LIST OF FIGURES

Figure 1.1: Wireless Ad-hoc Network ................................................................. 1
Figure 1.2: Biomedical Sensor Network application ........................................... 5
Figure 1.3: On-Chip Network ........................................................................... 7
Figure 2.1: Sensor network topology consisting 12 nodes ................................ 19
Figure 2.2: Average temperature rise vs. packet arrival rate for 12 nodes topology 19
Figure 2.3: Average packet delivery delay vs. packet arrival rate for 12 nodes topology 21
Figure 2.4: Average percentage power consumption vs. packet arrival rate for 12 nodes topology 22
Figure 2.5: Total number of packets dropped vs. packet arrival rate for 12 nodes topology 23
Figure 2.6: 4x4 Regular Mesh topology ............................................................. 23
Figure 2.7: Average temperature rise vs. packet arrival rate for 4x4 Mesh topology 24
Figure 2.8: Average packet delivery delay vs. packet arrival rate for 4x4 Mesh topology 25
Figure 2.9: Average percentage power consumption vs. packet arrival rate for 4x4 Mesh topology 25
Figure 2.10: Total number of packets dropped vs. packet arrival rate for 4x4 Mesh topology 26
Figure 2.11: 4x4 Mesh-Torus topology ............................................................. 26
Figure 2.12: Average temperature rise vs. packet arrival rate for 4x4 Mesh-torus topology 27
Figure 2.13: Average packet delivery delay vs. packet arrival rate for 4x4 Mesh-torus topology 28
Figure 2.14: Average percentage power consumption vs. packet arrival rate for 4x4 Mesh-torus topology 28
Figure 2.15: Total number of packets dropped vs. packet arrival rate for 4x4 Mesh-torus topology 29
Figure 2.16: 8 nodes Ring topology ................................................................. 30
Figure 2.17: Average temperature rise vs. packet arrival rate for 8 nodes Ring topology 30
Figure 2.18: Average packet delivery delay vs. packet arrival rate for 8 nodes Ring topology 31
Figure 2.19: Average percentage power consumption vs. packet arrival rate for 8 nodes Ring topology 32
Figure 2.20: Total number of packets dropped vs. packet arrival rate in the network for 8 nodes Ring topology ................................................................. 33
Figure 2.21: Randomly connected topology of 50 densely connected nodes .................. 33
Figure 2.22: Average temperature rise vs. packet arrival rate for network of 50 randomly connected nodes ................................................................. 34
Figure 2.23: Average packet delivery delay vs. packet arrival rate for network of 50 randomly connected nodes ................................................................. 35
Figure 2.24: Average percentage power consumption vs. packet arrival rate for network of 50 randomly connected nodes ................................................................. 35
Figure 2.25: Total number of packets dropped vs. packet arrival rate for network of 50 randomly connected nodes ................................................................. 36
Figure 2.26: Lifetime of the networks using the four routing algorithms (When at least 30% of the nodes are alive) ................................................................. 37
Figure 2.27: Lifetime of the networks using the four routing algorithms (When all alive nodes remain connected) ................................................................. 38
Figure 3.1: Mesh-torus topology of 32 nodes .................................................. 46
Figure 3.2: Maximum temperature rise vs. Packet arrival rate using THRESH_TARA=1.00 in 4x8 Mesh-torus topology ................................................................. 47
Figure 3.3: Average packet delivery delay vs. Packet arrival rate using THRESH_TARA=1.00 in 4x8 Mesh-torus topology ................................................................. 48
Figure 3.4: Number of packets dropped vs. Packet arrival rate using THRESH_TARA=1.00 in 4x8 Mesh-torus topology ................................................................. 48
Figure 3.5: Maximum temperature rise of the sensor nodes using THRESH_TARA=7.50 in 4x8 Mesh-torus topology ................................................................. 49
Figure 3.6: Maximum temperature rise vs. Packet arrival rate using THRESH_TARA=7.50 in 4x8 Mesh-torus topology ................................................................. 50
Figure 3.7: Average packet delivery delay vs. Packet arrival rate using THRESH_TARA=7.50 in 4x8 Mesh-torus topology ................................................................. 50
Figure 3.8: Number of packets dropped vs. Packet arrival rate using THRESH_TARA=7.50 in 4x8 Mesh-torus topology ................................................................. 51
Figure 3.9: 3D topology of 27 nodes ................................................................. 52
Figure 4.1: 11x11 mesh topology with SINK at the center ............................................................... 72
Figure 4.2: Maximum temperature rise of nodes vs. Network load ........................................... 73
Figure 4.3: Average energy consumption per node vs. Network load ........................................ 74
Figure 4.4: Percentage packet delivery vs. Network load ............................................................. 75
Figure 4.5: Average packet delivery delay vs. Network load ......................................................... 76
Figure 4.6: Energy consumption of all the nodes in the 11x11 mesh topology using RAIN at high load... 77
Figure 4.7: Energy consumption of all the nodes in the 11x11 mesh topology using C-FLOOD at high load ................................................................. 78
Figure 4.8: Energy consumption of all the nodes in the 11x11 mesh topology using RAIN at low load.... 79
Figure 4.9: Energy consumption of all the nodes in the 11x11 mesh topology using C-FLOOD at low load ................................................................. 80
Figure 4.10: Maximum temperature rise of all the nodes in the 11x11 mesh topology at high load. The upper mesh is for C-FLOOD and the lower one is for RAIN ......................................................... 81
Figure 4.11: Maximum temperature rise of all the nodes in the 11x11 mesh topology at low load. The upper mesh is for C-FLOOD and the lower one is for RAIN ......................................................... 82
Figure 4.12: Topology consisting of 100 randomly connected nodes with SINK in the middle ........ 82
Figure 4.13: Maximum temperature rise of nodes vs. Network load ........................................... 83
Figure 4.14: Average energy consumption per node vs. Network load ........................................ 84
Figure 4.15: Percentage packet delivery vs. Network load ............................................................. 85
Figure 4.16: Average packet delivery delay vs. Network load ......................................................... 86
Figure 4.17: Death of the nodes in the 11x11 mesh topology when using RAIN and C-FLOOD under low network load ................................................................. 89
Figure 4.18: Death of the nodes in the 11x11 mesh topology when using RAIN and C-FLOOD under high network load ................................................................. 90
Figure 4.19: Operational life of the network before any node dies when using RAIN and C-FLOOD in different network topologies under low load ......................................................... 91
Figure 4.20: Operational life of the network before any node dies when using RAIN and C-FLOOD in different network topologies under high load ......................................................... 92
Figure 4.21: Operational life of network with more than 50 percent of the nodes being able to reach the SINK when using RAIN and C-FLOOD in different network topologies under low load ......................................................... 93
Figure 4.22: Operational life of network with more than 50 percent of the nodes being able to reach the SINK when using RAIN and C-FLOOD in different network topologies under high load ......................................................... 94
Figure 5.1: Biocomm- Cross-layer protocol stack design ............................................................ 97
Figure 5.2: Request To Send (RTS) Frame Format (Standard 802.11) .......................................... 98
Figure 5.3: Clear To Send (CTS) Frame Format (Standard 802.11) ................................................. 99
Figure 5.4: Regular mesh topology with SINK at the center .......................................................... 105
CHAPTER 1
INTRODUCTION

1.1. Wireless Ad-hoc Networks

In regions where there is no network infrastructure, communication is still possible among devices or nodes having wireless communication capability by formation of an ad-hoc network. An ad hoc network is an autonomous system of mobile or fixed hosts, connected by wireless links. Due to the limited transmission range of the nodes in the network, communication in such infrastructure-less networks occur by multiple hops. Data communication and routing in such networks is challenging due to the varying network dynamics. All the nodes might not remain static in an ad-hoc network and the types of devices connected together in the network might not be homogeneous and have different processing power and memory-capacity.

Figure 1.1: Wireless Ad-hoc Network
Ad hoc networking is a multi-layer problem. The physical layer should adapt to the dynamically changing link characteristics. The Medium Access Control (MAC) layer should operate efficiently in the presence of hidden and exposed nodes. The problems encountered in the network layer include topology control, data access and service control.

Topology control problems include discovering neighbors, identifying their position, determining transmission radius, establishing stable wireless links to neighbors, scheduling node sleep and active periods, clustering and maintaining the selected structure. Data communication problems include routing, broadcasting, multicasting and geocasting. Service access problems include Internet access, cellular network access, data or service replication upon detection or expectation of network partition.

1.2. Wireless Sensor Networks

The emerging field of wireless sensor networks combines sensing, computation, and communication into a single tiny device. Through advanced networking protocols, these devices form a sea of connectivity that extends the reach of cyberspace out into the physical world. Although the capabilities of these tiny devices or sensor “motes” are minimal, deployment of a large number of such motes offers immense technological possibilities.

The development of sensor networks needs research in hardware, software and algorithms and thus encompasses three different research areas namely sensing, communication and computing. Current sensor networks can exploit technologies that were not available 20 years back and so applications of sensor networks that were not even dreamt of at that time are now being deployed. MEMS technology, more reliable
wireless communication, and low-cost manufacturing have resulted in small, inexpensive, and powerful sensors motes with embedded processing and wireless networking capabilities.

Current and potential applications of wireless sensor networks include intrusion detection, security and tactical surveillance, disaster management, weather monitoring, habitat monitoring applications, traffic surveillance, building and structures monitoring and monitoring of different biological parameters inside the human body. Use of networks of wireless motes in different applications requires the networks to have different application-specific performance requirements. Some sensor network applications like intrusion detection, security and tactical surveillance might be highly delay sensitive, while some others like long time monitoring applications might tolerate some communication delay but need the network to have a long operational life.

1.2.1. Constraints of Wireless Sensor Networks

Owing to their extremely small size and limited resources, sensor networks have a large number of operational constraints. Since wireless sensor nodes are powered by small built-in batteries, networks of such nodes are highly energy constrained as frequent recharging of the batteries is always not feasible. Often such sensor nodes are deployed in thousands from an airplane over the area of interest. In such cases removing the motes from the site to replace their batteries is impossible. However in networks of in-vivo sensor nodes since the nodes are extremely small in size, they need not be always removed from their operation site and recharged as they can be easily recharged by external IR radiation. But even then frequent recharging is not desired. So power consumption of the motes is a constraint in sensor networks deployed for any application.
If the nodes become depleted of power very rapidly, the multi-hop network formed by the sensor nodes will get disconnected and the network will fail to perform the desired operations. Hence any protocol designed for use in sensor network must aim at reducing the overall power consumption of the nodes in the network.

Due to the power constrains and small size, the radio antenna contained on the mote needs to by small in size and having a very low power consumption. Studies have shown that the energy expended by radio communication varies as $E=r^4$ where $r$ is the transmission radius of the wireless radio. So the communication range of the motes cannot be very large due to energy constraints. Thus in a wide-scale deployment of such sensor motes the data needs to propagate to the Base-Station through multiple small hops instead of a single large hop. So the routing algorithm used in such networks is very critical as it has to decide the optimum path that the data packets should take to be delivered to the destination, keeping in mind all the constraints of the sensor motes.

The wireless sensor motes are also highly constrained by their processing power and memory capacity. So they cannot run very complex communication algorithms. Also distributed algorithms are preferred as centralized algorithms usually involve a lot of messaging overhead, which causes unnecessary power drain.

Apart from these constraints, which are common to all types of sensor networks, sensor networks deployed for a specific application can have additional constraints due to different application performance requirements, deployment scenario etc.

1.2.2. Biomedical Sensor Networks

One of the most promising application areas of Wireless Sensor Networks in recent years has been in the field of Bio-medical research. Smart embedded biomedical
sensors have the potential to bring a radical change in medical diagnosis. Tiny wireless sensor motes developed using MEMS technology can be implanted inside the human body for monitoring different organs or biometrics [1-3], [6-8]. These motes, having radio communication capabilities collect different biometric data like temperature, glucose level, concentration of certain minerals and gases in blood and communicate the data wirelessly to a Base-station, usually located outside the body through a multi-hop wireless network. The Base-station usually has much higher computation capabilities and can analyze the raw data to come up with critical conclusions.

Since such networks of embedded sensors communicate wirelessly, they can be used in artificial retina, glucose level monitors, organ monitors, cancer detectors and a multitude of other medical applications [6], [11].

The development of in-vivo smart sensors to remedy medical problems has a lot of benefits to individuals and the society as a whole [1-3], [6], [11]. Once the technology of developing such in-vivo sensor networks gets more advanced, the diseases like cancer,
for which the early detection is the key to remedy, will be detected quickly and thus the medical costs for correcting chronic medical conditions will be greatly reduced [6].

1.2.3. Habitat-monitoring Sensor Networks

Wireless Sensor Networks can be used for different long term and short term monitoring applications. Habitat and environmental monitoring represent a class of sensor network applications that has a lot of benefits to different scientific community. Researchers in the Life Sciences are becoming increasingly concerned about the potential negative impacts of human presence in monitoring plants and animals in field conditions. It is almost certain that chronic human disturbance distort results of the different studies by changing behavioral patterns or distributions of the plants or animals being studied. Anthropogenic disturbance can also seriously reduce or even destroy populations of sensitive species by increasing stress, reducing breeding success, increasing predation, or causing a shift to unsuitable habitats. While the effects of disturbance are usually immediately obvious in animals, plant populations are sensitive to such disturbances. Disturbance effects are of particular concern in small island situations. Seabird colonies are notorious for their sensitivity to human disturbance. Research in Maine suggests that even a 15 minute visit to a cormorant colony can result in up to 20% mortality among eggs and chicks in a given breeding year. Repeated disturbance will lead to complete abandonment of the colony. On Kent Island, Nova Scotia, researchers found that Leach’s Storm Petrels are likely to desert their nesting burrows if they are disturbed during the first 2 weeks of incubation.

Wireless Sensor Networks represent a significant advance over traditional invasive methods of monitoring. Sensors can be deployed prior to the onset of the
breeding season or other sensitive period. Sensor network deployment may represent a substantially more economical method for conducting long-term studies than traditional personnel-rich methods.

The main constraint that the habitat monitoring sensor networks have is that the network must have a very long operational life and hence the protocols used in the network must be energy-efficient. The network must also have minimal effect on the microclimate of the habitat or nest of the species being studied.

1.2.4. Networks On-Chip

The advances in silicon technology and VLSI have revolutionized the electronics industry. Due to the ever increasing miniaturization of transistors, very complex chip designs are becoming possible. For both physical and complexity reasons, future chip designs will inherently be multiprocessor systems, consisting of hundreds of modules.
The System-on-Chip (SoC) designs have provided solution to complex problems of telecommunication and consumer electronics industry [40-41]. However traditional buses and interconnects have become the bottleneck in the advancement and growth of future SoCs [18-19], [40]. Studies have shown that buses cannot scale beyond a certain number of components on the chip [18-19]. Design of large-scale chips needs to be structured resulting in the decoupling of the communication from computation. So the solution to this scalability problem is to use a Network-on-Chip (NoC) that connects the different components on the chip by a packet switched network [18-19], [40]. They help decoupling computation and communication and offer well defined interfaces.

1.3. Routing Algorithms

In a packet switched network, the routing protocol decides the paths that packets should take to go from a source node to a destination node. The routes chosen for delivering packets can greatly affect the formation of hotspots in the network.

Routing in sensor networks is very challenging due to several characteristics that distinguish them from contemporary communication and wireless ad hoc networks [17]. First, due to the relatively large number of sensor nodes, it is not possible to build a global addressing scheme for the deployment of a large number of sensor nodes as the overhead of ID maintenance is high. Thus, traditional IP-based protocols may not be applied to WSNs. In WSNs, sometimes getting the data is more important than knowing the IDs of which nodes sent the data. Second, in contrast to typical communication networks, almost all applications of sensor networks require the flow of sensed data from multiple sources to a particular BS. Third, sensor nodes are tightly constrained in terms of
energy, processing, and storage capacities. Thus, they require careful resource management.

The design of routing protocols in WSNs is influenced by many challenging factors. These factors must be overcome before efficient communication can be achieved in WSNs. In the following, we summarize some of the routing challenges and design issues that affect routing process in WSNs.

**Node deployment**: Node deployment in WSNs is application dependent and affects the network topology and hence the performance of the routing protocol. The deployment can be either deterministic or randomized. In deterministic deployment, the sensors are placed at predetermined desired locations and data is routed through predetermined paths. However, in random node deployment, the sensor nodes are scattered randomly creating an infrastructure in an ad hoc manner. In a multi-hop WSN, each node plays a dual role as data source and data router. The malfunctioning of some sensor can cause significant topological changes and might require rerouting of packets.

**Fault Tolerance**: Some sensor nodes may fail or be non-operational due to lack of power, physical damage, or environmental interference. The failure of sensor nodes should not affect the overall task of the sensor network. If nodes fail, the routing protocols must accommodate formation of new paths to route data. Therefore some bounded redundancy may be needed in a fault-tolerant sensor network.

**Scalability**: The number of sensor nodes deployed in the sensing area may be in the order of hundreds or thousands, or more. Any routing scheme must be able to work with this huge number of sensor nodes.
Connectivity: High node density in sensor networks prevents them from being completely isolated from each other. Therefore, sensor nodes are expected to be highly connected. This, however, may not prevent the network topology from being variable and the network size from being shrinking due to sensor node failures. In addition, connectivity depends on the, possibly random, distribution of nodes.

Coverage: In WSNs, each sensor node obtains a certain view of the environment. A given sensor's view of the environment is limited both in range and in accuracy; it can only cover a limited physical area of the environment. Hence, area coverage is also an important design parameter in WSNs.

Data Aggregation: Since sensor nodes may generate a fairly large amount of redundant data. Similar packets from multiple nodes can be aggregated so that the number of transmissions is reduced, hence saving energy. This technique has been used to achieve energy efficiency and data transfer optimization in a number of routing protocols.

Quality of Service: In some applications, data should be delivered within a certain period of time from the moment it is sensed; otherwise the data will be useless. Therefore bounded latency for data delivery is a necessary criterion for time-constrained or delay-sensitive applications. However, in many applications, conservation of energy, which is directly related to network lifetime, is considered relatively more important than the quality of data sent.

Routing protocols must select the best path to minimize the total power needed to route packets in the network and to maximize the lifetime of the network. That is, these protocols should be scalable to obtain different energy and quality operating points, as the
relative importance of different resources might change over the system lifetime. Also, wireless sensor networks need protocols which are data centric, capable of effective data aggregation, distribute energy dissipation evenly, efficiently use their limited energy to increase the longevity of the network and avoid any single point bottleneck (except the sink).

A very large number of routing algorithms have been developed for wireless sensor networks. They deal with various design constraints like load balancing, energy efficiency and delay constraints.

One of the most conventional approaches in the design of a routing protocol for sensor networks is based on conserving the node’s energy. Thus many power aware routing protocols have been proposed [12], [42-44]. QoS-aware protocols on the other hand consider end-to-end delay requirements [45-46].

1.4. Medium Access Control Protocols

The Medium Access Control or MAC layer is a sub-layer of the data link layer in the network protocol stack. Medium access control (MAC) protocols have been developed to assist each node to decide when and how to access the channel. This problem is also known as channel allocation or multiple access problem. MAC protocols have been extensively studied in traditional areas of wireless voice and data communications. Time division multiple access (TDMA), frequency division multiple access (FDMA) and code division multiple access (CDMA) are MAC protocols that are widely used in modern cellular networks. Their basic idea is to avoid interference by scheduling nodes onto different sub-channels, which are divided either by time, frequency or orthogonal codes. Since these sub-channels do not interfere with each other,
MAC protocols in this group are largely collision-free. We refer to them as scheduled protocols.

Another class of MAC protocols is based on contention. Rather than pre-allocate transmissions, nodes compete for a shared channel, resulting in probabilistic coordination. Collision happens during the contention procedure in such systems. Classical examples of contention-based MAC protocols include ALOHA and carrier sense multiple access (CSMA). In ALOHA, a node simply transmits a packet when it is generated (pure ALOHA) or at the next available slot (slotted ALOHA). Packets that collide are discarded and will be retransmitted later.

In CSMA, a node listens to the channel before transmitting. If it detects a busy channel, it delays access and retries later. The CSMA protocol has been widely studied and extended; today it is the basis of several widely-used standards including IEEE 802.11.

Sensor networks differ from traditional wireless voice or data networks in several ways. Sensor nodes conserve energy by turning off unneeded hardware because most hardware, even when not active, consumes a non-negligible amount of energy. Thus, each sensor node must somehow coordinate with its neighbors to ensure both devices remain active and participate in communication. Other design constraints, such as fairness, latency, and throughput, appear for specific applications and we present MAC protocols designed with these constraints.

1.5. Cross-Layer Protocol Design

Most protocols designed for wireless networks follow the traditional layered structure of the Open System Interconnection (OSI) model. While the protocols
developed for a particular layer achieve high performance in terms of the metrics related to each of these individual layers, they are not jointly optimized so as to maximize the overall network performance.

In wireless sensor networks traditional layer by layer protocol design might be very inefficient due to the scarcity of energy and processing resources. So joint design and optimization of the layers of the protocol stack are needed. Some recent works reveal that cross-layer design and integration techniques result in significant reduction in energy consumption of the WSNs. In cross-layer design the different layers interact very closely to achieve high performance goals.
CHAPTER 2
ENERGY EFFICIENT THERMAL AWARE ROUTING
ALGORITHMS FOR EMBEDDED BIOMEDICAL SENSOR
NETWORKS

2.1. Introduction

Existing communication protocols may not be appropriate for use in networks of biomedical sensors since such networks impose some additional constraints on the protocols [6-9]. Since wireless sensor nodes are powered by small batteries, networks of such nodes are highly energy constrained as frequent recharging of the batteries is always not feasible. Since the in-vivo sensor nodes are extremely small in size, they need not be always removed from their operation site and recharged as they can be easily recharged by IR radiation [7-9]. But even then, frequent recharging is not desirable; power consumption is an important constraint in embedded biomedical sensors like in most other sensor network applications as reliability of the embedded network is very important for bio-medical applications. If the nodes become depleted of power very rapidly the multi-hop sensor network formed by the nodes will get disconnected and the network will fail to perform the desired operations.

The tiny in-vivo sensor nodes are also constrained by their memory capacity and processing power. So the routing algorithm used by the nodes should not involve very complex routing logic. Also distributed algorithms are preferred as centralized algorithms usually involve a lot of messaging overhead.

Communication radiation and power dissipation of the implanted sensor nodes can cause serious health hazards [6-7], [10]. A high temperature of the embedded nodes for a very long time might damage the surrounding tissues. The heat produced by the
sensor nodes might also foster the growth of certain bacteria that would otherwise be absent. It might also have a subtle effect on the enzymatic reactions inside the body [6]. So the communication protocols must aim at reducing the temperature rise of the nodes in the in-vivo sensor network.

To our knowledge, the only routing algorithm that considers the heat produced due to communication as a constraint is the Thermal Aware Routing Algorithm or TARA [7]. TARA routes the data away from high temperature areas (hot spots). Basically, each node listens to its neighbors’ activity, counting the packets they transmitted and received. The node can then evaluate the communication radiation and power dissipation of its neighbors, and estimate their temperature change. Once the temperature of one neighbor exceeds a predefined threshold, the node would mark that neighbor as a hot spot and would route packets around the hot spot area by a withdrawal strategy. The essence of this withdrawal strategy is as follows. When a node gets a packet and all of its outgoing neighbors are hot spots, it returns the packet to the sender (previous node). The previous node would try to send the packet to an alternative node or may send it to the previous node. In summary, TARA tries to solve the problem of tissue heating by a strategy that withdraws packets from heated zones and route them through alternate paths.

There are certain delay-sensitive medical applications where a large delay in communicating the sensed information to the base-station might be fatal. Certain other real-time applications need to transmit the sensed data with minimum delay to the base-station e.g. the sensor network deployed in artificial retina [11]. So the communication protocols also need to minimize the delay in communicating the sensed information to the base-station.

In this chapter we propose two routing algorithms for embedded biomedical sensors, Least Temperature Routing (LTR) and Adaptive Least Temperature Routing
(ALTR) [1], and compare their performance with TARA and the Shortest Hop Routing (SHR) algorithm.

2.2. Proposed Algorithms

2.2.1. Least Temperature Routing Protocol

- **Setup Phase**: In the setup phase, the nodes communicate with the neighbors and gather information about their temperature.

- **Routing**: A node has to handle two types of packets, those that originate due to sensing events by the node itself and those that are coming from other nodes [6]. Both types of packets are treated in the same way. If the destination node of the packet is a neighboring node, the packet is forwarded directly to its destination node. As in TARA [7], we assume each node has information about the temperature of its neighbors. The node forwards the packet to the neighbor having the least temperature or the “coolest neighbor” if the destination node is not one of its neighbors.

- **Packet discard**: Each packet has a hop-count which is incremented by one each time a node forwards a packet. Once a node gets a packet, it checks the hop-count. If it is above a certain threshold value MAX_HOPS, the packet is discarded. The value of MAX_HOPS depends on the diameter of the sensor network.

- **Reducing unnecessary hops and loops**: Each packet maintains a small list of nodes it has most recently visited. If the “coolest neighbor” is already there in the list of recently visited nodes for the packet, the packet is forwarded to the neighbor having the second lowest temperature among the neighboring nodes. Similarly, if the “coolest neighbor” is a leaf node and is not the final destination of the packet, the packet is forwarded to the neighbor having the next lower temperature among the
neighboring nodes. The list of recently visited nodes should include all nodes visited within some past window. Although it is better to have large window, the size of the list is practically constrained and must be kept small in order to prevent packets from becoming too large as well as reduce the overhead of searching the list.

2.2.2. Adaptive Least Temperature Routing protocol

The Adaptive Least Temperature Routing (ALTR) protocol is similar to the LTR protocol with some improvements made to minimize the packet delivery delay and have good performance in different network topologies.

- **Minimizing delay:** In this protocol, a node having a packet to route checks the hop-count of the packet. If the hop-count is less than or equal to a threshold value MAX_HOPS_ADAPTIVE, the packet is routed following the rules of the LTR algorithm. If the hop-count is greater than MAX_HOPS_ADAPTIVE, the packet is routed using the shortest hop (SHR) algorithm. MAX_HOPS_ADAPTIVE is a value that is set by experiment.

- **Adapting to different topologies:** In certain topologies where the degree of connectivity is low, e.g. a ring topology, although the delay remains very low, the temperature of the nodes increases rapidly as successive packets get routed along the same path repeatedly. To solve this problem, the ALTR scheme uses a “proactive delay” mechanism. If a node receives a packet and has no more than two (outgoing) neighbors such that its coolest neighbor has a relatively high temperature, the node delays the packet by one time unit before forwarding it to the coolest neighbor. In this way, the average temperature of the sensor network is reduced at the expense of a slight increase in the packet delivery delay.
2.3. Simulation

The simulation program was developed in C. The nodes in our simulation sense data and route the information to a base station through the multi-hop wireless network. We performed extensive simulation tests over the following network topologies: i) a randomly connected 12 nodes topology, ii) a 4x4 mesh topology, iii) a 4x4 mesh-torus topology, iv) a 8 nodes ring topology and v) a network of 50 randomly connected nodes to verify the scalability of the proposed routing protocols.

We compared the performance of the proposed protocols Least Temperature Routing (LTR) and Adaptive Least Temperature Routing (ALTR) with the Shortest Hop Routing (SHR) protocol and the Thermal Aware Routing Algorithm (TARA) proposed in [7] with increasing packet arrival rate or packet injection rate in the system.

The metrics used for the comparison are average temperature rise, the average delay in packet delivery, the average power consumption of the sensor nodes and the total number of packets dropped in the network. The average temperature rise of the nodes gives the average change in temperature of the nodes from that at the beginning of the simulation.

Each simulation run generated 2000 data packets that are routed through the network. Each time a node receives a packet to route, its temperature increases by 1 unit. The cooling rate of the system (all the nodes) is assumed to be 1 unit per simulation interval.

We assume that idle listening consumes one unit of power. Both transmission and reception of each packet by a node consume two power units [12]. In all of the algorithms, it has been assumed that the maximum number of hops allowed is equal to
40. The ALTR follows LTR till 10 hops (the value of MAX_HOPS_ADAPTIVE) and then switches to the SHR algorithm.

Experiments have also been performed to determine the lifetime of the network for each of the four routing algorithms in the different topologies.

2.3.1. Network of 12 randomly connected nodes

![Sensor network topology consisting 12 nodes](image)

**Figure 2.1: Sensor network topology consisting 12 nodes**

![Average temperature rise vs. packet arrival rate for 12 nodes topology](image)

**Figure 2.2: Average temperature rise vs. packet arrival rate for 12 nodes topology**
Average temperature rise of the nodes in routing the data packets gives an estimate of the amount of heat produced in the network by the nodes due to the wireless communication. Figure 2.2 shows the average temperature rise in the system with increasing packet arrival rate. It can be observed that the LTR and ALTR produce much lower average temperature rise than SHR and TARA at low packet arrival rates. At higher packet arrival rates, both LTR and ALTR produce higher average temperature rise but their rise is still much lower than that of SHR and slightly lower than TARA. At high packet arrival rates, TARA performs better than SHR but LTR and ALTR perform the best.

The results in Figure 2.2 can be explained as follows. At higher packet arrival rates, TARA tries to withdraw the packets from the vicinity of the heated nodes and tries to route the packet through alternate paths. Thus in effect it increases the amount of communication in the network and hence the average temperature of the network. SHR tries to route the packets following the shortest path, disregarding the temperature rise and thus raising the temperature even higher. LTR and ALTR try to route the packets through the ‘cooler’ nodes from the beginning and thus perform well even at high packet arrival rates.
Delay experienced in delivering a packet is an important factor for biomedical sensor network applications that are generally time critical. Figure 2.3 shows the average delay experienced in delivering a packet vs. the packet arrival rate. The SHR always follows the shortest path in routing a packet. So the average delay experienced by packets is the least in case of SHR. TARA has low delay at lower packet arrival rates and the delay increases considerably with the increase in packet arrival rate. The LTR and ALTR produce almost the same amount of delay at all packet arrival rates. At higher packet arrival rates the LTR and ALTR algorithms perform much better than TARA.

The results can be explained as follows. At higher packet arrival rates, when the average temperature of the network is high, TARA tries to withdraw the packets from the heated areas and reroute them through alternate paths. Thus TARA experiences higher delay due to rerouting the packets at higher packet arrival rates. LTR and ALTR try to
route the packets through the cooler part of the network, irrespective of the packet arrival rate. Thus the packet delivery delay remains almost the same.

Figure 2.4: Average percentage power consumption vs. packet arrival rate for 12 nodes topology

Power consumption of the nodes is an important factor to be considered in analyzing the performance of any algorithm for sensor networks. It is evident from Figure 2.4 that the SHR has the least power consumption as the packets are routed quickly through the shortest path and thus minimizing communication. Both LTR and ALTR have slightly higher power consumption than SHR but their consumption is much lower than that of TARA for all packet arrival rates. The power consumption of the nodes using TARA rises considerably with the increase in packet arrival rate. TARA in an attempt to withdraw the packets from the heated area and reroute them incur higher power consumption.
From Figure 2.5, it can be seen that no packets are dropped or very few packets are dropped when using SHR, LTR and ALTR. TARA, however, drops a large number of packets at higher packet arrival rates. In an effort to reroute the packets away from the heated zones by TARA, the packets experience large number of hops and a large number of packets are dropped.

2.3.2. 4x4 Mesh topology

Figure 2.6: 4x4 Regular Mesh topology
In Figure 2.7, the average temperature rise in the 4x4 mesh network has similar characteristics as that of the 12-node network of Figure 2.2 except the fact that in this topology the performance of TARA deteriorates faster than it does in the 12 node network topology. In this topology also the performance of LTR and ALTR are the best.

In Figure 2.8, for a mesh topology TARA, LTR and ALTR show lower delay than when they are used in the 12-node network. This is due to the fact that in the mesh network there are many alternative paths between two nodes. In the 4x4 mesh topology, the ALTR has significantly lower delay than the LTR. This is due to the adaptive nature of ALTR by which it takes the shortest path to the destination if the packet hop-count goes above a particular threshold.

Figure 2.7: Average temperature rise vs. packet arrival rate for 4x4 Mesh topology
Figure 2.8: Average packet delivery delay vs. packet arrival rate for 4x4 Mesh topology

Figure 2.9: Average percentage power consumption vs. packet arrival rate for 4x4 Mesh topology

From Figure 2.9, it can be observed that with respect to power consumption, all four algorithms perform similar to the scenario when the algorithms were run over the
12-node network. However, TARA performs better than it does in case of the 12 node irregular topology.

Figure 2.10: Total number of packets dropped vs. packet arrival rate for 4x4 Mesh topology

From Figure 2.10 it can be seen that with respect to total number of packets dropped, all four algorithms perform similar to the scenario when the algorithms were run over the 12-node network.

2.3.3. 4x4 Mesh Torus network

Figure 2.11: 4x4 Mesh-Torus topology
As shown in Figure 2.12, in a Mesh-torus topology all of the four algorithms produce a very small rise in temperature till arrival rate of 1.5. The SHR algorithm produces a slow rise in the average temperature after that while the average temperature rise of LTR and ALTR continues to remain low. TARA produces a steep rise in temperature at arrival rate of 1.75. As will be explained at the end of this section, this steep rise is due to packets continuously looping in the mesh-torus topology.

All the four algorithms produce low delay till arrival rate of 1.5 as shown in Figure 2.13. SHR, LTR and ALTR continue with low delay. TARA produces a steep rise in delay at arrival rate of 1.75.
Figure 2.13: Average packet delivery delay vs. packet arrival rate for 4x4 Mesh-torus topology

Figure 2.14: Average percentage power consumption vs. packet arrival rate for 4x4 Mesh-torus topology
From Figure 2.14, it can be seen that the percentage power consumed remains nearly constant throughout for SHR, LTR and ALTR. Till an arrival rate of 1.5, TARA and SHR have the lowest power consumption. LTR and ALTR have slightly higher power consumption. At arrival rate of 1.75 TARA shows a sharp increase in power consumption.

All of the four algorithms produce a very low packet drop till arrival rate of 1.50 as shown in Figure 2.15. The packet dropping of SHR, LTR and ALTR continue to remain low. TARA produces a steep rise in packet drop at arrival rate of 1.75.

From Figures 2.12, 2.13, 2.14 and 2.15 it can be seen that the performance of TARA in general deteriorates sharply at arrival rate of 1.75. This is due to the fact that TARA withdraws packets from hotspots and routes them through alternate paths and the mesh torus topology provides a lot of alternate paths between two nodes. So the packets continue to loop in the network, as TARA does not prevent the packets from revisiting an
already visited node. This unnecessarily increases communication load and degrades performance. The sudden change in performance can be due to an abrupt increase in number of hotspots in the network.

2.3.4. 8-node Ring topology

![8-node Ring topology diagram](image)

Figure 2.16: 8 nodes Ring topology

![Average temperature rise vs. packet arrival rate for 8 nodes Ring topology](image)

Figure 2.17: Average temperature rise vs. packet arrival rate for 8 nodes Ring topology

As shown in Figure 2.17, the average temperature rise of the nodes in the ring topology has different characteristics from that observed in the other topologies. This is
because in a ring each node has only one outgoing neighbor for any incoming packet. So the options for forwarding a packet become extremely limited. In the ring topology, ALTR performs best as it uses the “proactive delay” mechanism. This mechanism helps the ALTR adapt to different topologies easily. The LTR does not perform good as the packets repeatedly follow the same path and it does not have any mechanism to adapt to the topology. The TARA does not perform as well as ALTR but performs better than SHR and LTR.

![DELAY WITH ARR RATE (8 NODES RING)](image)

**Figure 2.18: Average packet delivery delay vs. packet arrival rate for 8 nodes Ring topology**

In Figure 2.18, for a ring topology TARA shows a steep increase in delay when the packet arrival rate becomes greater than 1.00. SHR and LTR both have negligible delay. ALTR has low delay but it is still higher than that of SHR and LTR. This is due to the “proactive delay” mechanism used by ALTR to reduce the average temperature of the nodes.
Figure 2.19: Average percentage power consumption vs. packet arrival rate for 8 nodes Ring topology

It can be seen from Figure 2.19 that SHR has the least power consumption for the ring topology. Both LTR and ALTR have consistent relatively low power consumption but higher than SHR. TARA has low power consumption till a packet arrival rate of 1.00. After that, the power consumption of TARA increases sharply.

Figure 2.20 shows that SHR, LTR and ALTR do not have any packet drops. TARA shows a high packet drop at packet arrival rates greater than 1.00.
Figure 2.20: Total number of packets dropped vs. packet arrival rate in the network for 8 nodes Ring topology

2.3.5. Dense network of 50 randomly connected nodes

Figure 2.21: Randomly connected topology of 50 densely connected nodes
Figure 2.22: Average temperature rise vs. packet arrival rate for network of 50 randomly connected nodes

For the 50 nodes network the LTR and ALTR produce much lower average temperature rise than SHR and lower average temperature rise than TARA at all packet arrival rates as shown in Figure 2.22. TARA performs reasonably better than SHR but LTR and ALTR perform the best.

The SHR always follows the shortest path in routing a packet. So the average delay experienced by packets is the least in case of SHR as shown in Figure 2.23. The TARA, LTR and ALTR produce almost the same amount of delay at all packet arrival rates. However both the LTR and ALTR produce lower delays than TARA.
SHR, LTR and ALTR all have very low power consumption as shown in Figure 2.24. TARA has much higher power consumption than the other three protocols.

**Figure 2.23:** Average packet delivery delay vs. packet arrival rate for network of 50 randomly connected nodes

**Figure 2.24:** Average percentage power consumption vs. packet arrival rate for network of 50 randomly connected nodes
For the large 50 nodes network SHR, LTR and ALTR all have very little packet drops as shown in Figure 2.25. TARA has much higher packet drop than the other three protocols.

2.3.6. Lifetime of Network

The lifetime of the sensor network is a very important factor when evaluating the design of any protocol for sensor networks. The lifetime of the networks using the four different routing protocols has been compared. Lifetime has been defined in two ways. In the first case a network has been considered alive as long as 30% of all the nodes remain alive. In the second case a network has been considered alive as long as all the alive nodes remain connected.
Figure 2.26: Lifetime of the networks using the four routing algorithms (When at least 30% of the nodes are alive)

It can be seen from Figure 2.26 that the network has longest lifetime for all topologies in the first case when the SHR algorithm is used. The network lifetime is lowest when using TARA. This is due to the fact that the path chosen by the SHR for routing is the shortest and thus it drains the power of only a few nodes. TARA on the other hand withdraws packets from regions of high temperature and reroutes packets and thus draining a large number of nodes, resulting in reduced network lifetime. Networks using LTR and ALTR have longer lifetime than TARA.

In the second case (Figure 2.27) the results are similar to those in the first except the fact that when the algorithms are run on the 50-node network the LTR and ALTR give the network the highest lifetime.
Figure 2.27: Lifetime of the networks using the four routing algorithms (When all alive nodes remain connected)

2.4. Conclusion

This chapter presents two thermal aware routing protocols, LTR and ALTR. Both algorithms reduce the average temperature rise of the sensor nodes, the average packet delivery delay, the number of packets dropped and the average power consumption of the nodes. In certain topologies the ALTR adapts itself to both varying network load and the network topology and performs better than LTR. Both LTR and ALTR give the network longer life than TARA. Thus the network of embedded biomedical sensors using LTR and ALTR will continue to route data packets in an energy efficient way for longer time without rising the temperature of the network too much. The LTR and ALTR routing algorithms pave the way towards development of new upcoming applications of wireless sensor network in the field of medical science.
CHAPTER 3
HOTSPOT PREVENTING ROUTING ALGORITHM FOR DELAY-SENSITIVE APPLICATIONS OF IN-VIVO BIOMEDICAL SENSOR NETWORKS

3.1. Introduction

The Adaptive Least Temperature Routing (ALTR) algorithm [1] presented in Chapter 2 aims at reducing the total amount of heat produced in the network of biomedical sensor nodes and does not address the issue of formation of hotspots in the network. ALTR tries to reduce the average temperature rise of the sensor nodes. However, reducing the average temperature of the in-vivo nodes does not ensure that the network will not have any node having a very high temperature. The ALTR algorithm performs better than TARA [7] as far as minimization of average packet delivery delay is concerned only in some specific topologies and under specific network load conditions, as it does not use any explicit delay control mechanism till the hop-count of the packets reaches a preset threshold value.

In this chapter, we propose the Hotspot Preventing Routing (HPR) algorithm [2-3], that prevents the formation of hotspots in the network and is more suitable for being used in embedded biomedical sensor networks than ALTR algorithm, and compare its performance with TARA [7] and the Shortest Hop Routing (SHR) algorithm. Also unlike ALTR, HPR prevents the packets from taking unnecessarily long paths and performs better in terms of reducing the average packet delivery delay.

The performance of HPR has been compared with that of the Shortest Hop Routing (SHR) algorithm although SHR is not a thermal-aware routing algorithm. The reason for this is that the SHR algorithm always delivers packets to the sink through the
shortest paths and thus the average packet delivery delay is minimized. The proposed HPR algorithm has been developed for delay-sensitive applications of biomedical sensor networks. So while preventing the formation of hotspots in the network, the HPR algorithm also tries to match the performance of SHR in terms of minimizing the average packet delivery delay.

3.2. Proposed Algorithm

The Hotspot Preventing Routing (HPR) algorithm consists of two main phases, the setup phase and the routing phase.

- **Setup Phase:**
  - The nodes exchange information about the shortest path and build the routing tables. The nodes also exchange information about their initial temperatures with the neighbors.

- **Routing Phase:**
  - Each sensor node handles two types of packets, those that originate at the node itself due to sensing and those that are forwarded by other nodes to be routed to destination. Both types of packets are treated in the same way by the node.
  - Each node tries to route the packets in such a way that the packets get delivered to the destination using the minimum number of hops, i.e. the shortest path, unless there is a hotspot in the path.
  - If the destination of the packet is a neighboring node, the packet is forwarded directly to the destination.
o Else if the next-hop in the shortest path to the destination has a temperature less than or equal to the temperature of the node plus a \textit{threshold}, the packet is forwarded to the next-hop node in the shortest path to the destination.

o Else if the next hop in the shortest path to the destination has a temperature greater than the temperature of the current node plus a \textit{threshold}:
  
  - The node predicts that the path under consideration goes through a hotspot and tries to bypass the hotspot and route the packet.
  - The node forwards the packet to the neighbor node that has not been visited and has the least temperature or is the \textit{coolest neighbor}.
  - To prevent routing loops each packet maintains a small list of nodes it has most recently visited. If the \textit{coolest neighbor} is already there in the list of recently visited nodes for the packet, the packet is forwarded to the neighbor having second lowest temperature among all the neighboring nodes and so on.

o Experiments show that with the increase in the packet injection rate in the system, the average temperature of the nodes increases. So does the variation in temperature among the neighboring nodes. The value of the \textit{threshold} is dynamically set based on the local load (packets routed by the node over a past time window) and the temperature of the neighboring nodes. Both factors are given equal weight in calculating the \textit{threshold} value.

o The temperature of a node gives a measure of the number of packets routed by it over a past window and hence the load handled. Hence the \textit{threshold} value depends on two components:
- Component C1: A function of the average temperature of the neighboring nodes.
- Component C2: A function of the node’s own temperature.

  - C1 and C2 are found out by experiments.
    - \( C1 = k1 \times \sqrt{\text{avg}_n} \)
    - \( C2 = k2 \times \sqrt{\text{temp}_n} \)
    - \( k1 \) and \( k2 \) are constants set by experiments
    - \( \text{avg}_n \) is the average temperature of the node’s neighbors
    - \( \text{temp}_n \) is the temperature of the node.

  - The \textit{threshold} is calculated giving equal weights to the two components.
    \[ \text{threshold} = 0.5 \times C1 + 0.5 \times C2. \]

  - The nodes overhear the transmissions of the neighbors and based on that estimate the number of packets transmitted by the neighbor nodes within a certain interval of time. Based on that estimate, the nodes compute the change in temperature of the neighbor nodes in that time-interval. The sum of the change in temperature (may be positive or negative) of a neighbor and its previously computed temperature gives the neighbor’s current temperature. This helps the node maintain updated temperature values of the neighbors.

  - Each packet has a hop-count which is incremented by one each time a node forwards a packet.
o Once a node gets a packet, it checks the hop-count. If it is above a certain threshold value `MAX_HOPS`, the packet is discarded. The value of `MAX_HOPS` depends on the diameter of the sensor network.

### 3.3. Pseudo Code

In the pseudo-code provided, `Dequeue()` removes a packet from the node’s buffer queue for routing it. `Avg_Temp_N()` returns the average temperature of the neighbor nodes. `Send_Packet(P, N)` sends a packet `P` to node `N`. `Coolest_Neighbor(i)` returns the node-id of the i-th coolest neighbor. `Next-hop(D)` searches the routing table and returns the next hop the packet should take from the current node to reach node `D`. `P.Visited(N)` returns TRUE if and only if the packet `P` has visited node `N`.

**Procedure HPR()**

```plaintext
1: {
2:   Packet = Dequeue(Node.Buffer);
3:   /*Calculate dynamic Threshold*/
4:   C1=k1 * Sqrt(Avg_Temp_N());
5:   C2=k2 * Sqrt(Node.Temp);
6:   Threshold= 0.5* C1+0.5*C2
7:   If (Packet.Hop-count >= MAX_HOPS )
8:     Drop-Packet();
9:   Else    /*Packet needs to be routed*/
10:      {
11:         Packet.Hop-count++;
12:        /*Routing logic*/
```
13: If ( Packet.Destination == Neighbor N_i )
14:     Send_Packet ( Packet, N_i );
15: Else If ( Next-hop ( Packet.Destination ).Temp < ( Node.Temp + Threshold ) )
16:     Send_Packet ( Packet, Next-hop ( Packet.Destination ) );
17: Else If ( Next-hop ( Packet.Destination ).Temp>= ( Node.Temp + Threshold ) )
18:     {
19:         /* This prevents "Routing Loops" */
20:         Do {
21:             i = 1;
22:             N_c = Coolest_Neighbor ( i );
23:             i++;
24:         } While ( Packet.Visited ( N_c ) == TRUE && i < Node.Neighbors )
25:         Send_Packet ( Packet, N_c );
26:     }
27: }
28: }

3.4. Simulation

The simulation program has been developed in C. We performed the simulation over three different sensor network topologies and compared the performance of the proposed algorithm (HPR) with that of the SHR and TARA.

The metrics used for the comparison of the algorithms are:

A. Maximum temperature rise of any node in the network: The maximum temperature rise of any node in the network is an important metric for in-vivo
sensor networks that needs to be compared as it is directly associated with the formation of hotspots and the possibility of tissue damage. The average temperature rise of the nodes might not always be a suitable metric for comparison of the protocols, since a low average temperature rise of the in-vivo sensor nodes does not ensure that no hotspots are formed in the network.

B. Average packet delivery delay in the network: Average packet delivery delay is a measure of the average delay incurred to route a packet from a source to its destination. This is an important metric for delay sensitive and real-time biomedical applications.

C. Total number of packets dropped: The total number of packets dropped is an important metric in determining the performance of any routing algorithm because of the large overhead incurred in retransmitting dropped packets.

Each simulation run generated 8000 data packets that are routed through the network to a randomly chosen base station (The base station is fixed for each simulation run). The sources of the packets are randomly generated and packet arrival or packet injection in the network follows a Poisson distribution.

It has been assumed that each transmission or reception of a packet increases the temperature of the node and the surrounding tissues by 0.1 units. If a different value be used for the increase in temperature of the nodes, it will affect all the simulated protocols equally; so the relative difference in the performance of the simulated protocols will remain almost the same. Since tiny biomedical sensor nodes are still in the research and development phase, exact values of temperature rise due to communication among the in-vivo nodes is not known. However based on recent research on the temperature rises of
different parts of the body due to exposure to radio waves [13], [14], our assumption seems logical as once developed, the in-vivo wireless sensor nodes will most likely operate in the industrial, scientific and medical (ISM) radio bands [15].

The heat generated in the nodes due to communication among themselves gets dissipated with time, leading to a drop in the temperature of the nodes. The cooling rate of the system (for all the nodes) is assumed as 0.1 units per simulation interval. The rationale behind assuming this cooling rate is as follows. Since each transmission or reception increases the temperature of the corresponding node by 0.1 units, the cooling rate of the network should be at least 0.1 units per simulation interval for stable operation of the embedded network of sensor nodes.

Based on experiments the values of k1 and k2 are chosen to be 0.05. At this value the HPR algorithm gives overall optimum performance under all loads.

3.4.1. 4x8 Mesh-torus topology

One of the most commonly used topologies for sensor networks is the Mesh-torus topology as it avoids the boundary effects [16]. Figure 3.1 shows 32 sensor nodes connected in a 4x8 mesh-torus topology.
In the simulation initially the threshold value for TARA has been set equal to 1.00 (THRESH_TARA=1.00). This means that once the temperature rise of any neighbor exceeds 1.00, the node would mark that neighbor as a hot spot and would withdraw packets from it and reroute the packets. The results obtained are shown in Figures 3.2, 3.3 and 3.4.

![Graph](image.png)

**Figure 3.2: Maximum temperature rise vs. Packet arrival rate using THRESH_TARA=1.00 in 4x8 Mesh-torus topology**

From Figures 3.2, 3.3 and 3.4 it can be seen that in a mesh torus topology, although TARA seems to perform better than the other two algorithms when the packet arrival rate is high, it produces unacceptably high packet delivery delay and packet drop. The proposed HPR algorithm produces very low packet delivery delay and nearly zero packet drop. It also performs much better than SHR in terms of reducing the maximum temperature rise of the nodes.
Figure 3.3: Average packet delivery delay vs. Packet arrival rate using THRESH_TARA=1.00 in 4x8 Mesh-torus topology

Figure 3.4: Number of packets dropped vs. Packet arrival rate using THRESH_TARA=1.00 in 4x8 Mesh-torus topology
For comparing the protocols in an unbiased manner the value of the threshold used by TARA is set (THRESH_TARA=7.50), such that both TARA and HPR produce almost similar packet delivery delays at a moderately high packet arrival rate (2.5 packets per simulation interval). The results obtained are shown in Figures 3.5, 3.6, 3.7 and 3.8.

![Max Temp Rise in Nodes](image)

**Figure 3.5: Maximum temperature rise of the sensor nodes using THRESH_TARA=7.50 in 4x8 Mesh-torus topology**

Figure 3.5 compares the maximum temperature rise of the 32 nodes in the network when the three different algorithms are used for routing in the 4x8 mesh-torus topology. TARA reduces the maximum temperature rise of the hotspots by restricting the temperature value beyond the preset threshold value but HPR significantly reduces the temperature rise of the nodes and thus clearly performs the best.
Figure 3.6: Maximum temperature rise vs. Packet arrival rate using THRESH_TARA=7.50 in 4x8 Mesh-torus topology

Figure 3.7: Average packet delivery delay vs. Packet arrival rate using THRESH_TARA=7.50 in 4x8 Mesh-torus topology
Figures 3.6, 3.7 and 3.8 show that HPR performs much better than TARA and SHR at all packet arrival rates in terms of reducing the maximum temperature rise of the nodes. Although TARA produces lower temperature rise than the SHR, it produces unacceptably high packet delivery delay and packet drop at high packet arrival rates. Both HPR and SHR produce low packet delivery delay and nearly zero packet drop.

TARA tries to withdraw the packets from the vicinity of the heated nodes if the temperature of the node reaches a preset threshold and tries to route the packets through alternate paths. Thus in effect it increases the amount of communication in the network and hence the number of hotspots. The packet delivery delay also increases as any packet that has been withdrawn from the vicinity of a heated node follows an arbitrarily long path to the destination. Due to the same reason a large number of packets also get dropped.
SHR tries to route the packets following the shortest path and thus the packets always incur very low packet delivery delay and also none of the packets are dropped. But since the temperatures of the nodes in the path are disregarded when routing packets, hotspots having very high temperature are formed in the network.

HPR does not route packets through arbitrary long paths in the network. Instead it always tries to route packets through the shortest path from the source to the destination, only bypassing the zones of high temperature that appear in the path. When a packet approaches a hotspot it is forwarded to the relatively cooler part of the network. From there the packet is again forwarded following the shortest path to the destination. This prevents the packets from taking an arbitrarily long path and thus reduces the packet delivery delay and packet drops. HPR also dynamically adjusts the route based on the network traffic conditions. Thus HPR prevents formation of hotspots having extremely high temperature as well as delivers packets to the destination with a very low packet delivery delay.

3.4.2. Topology consisting of a 3D layout of 27 nodes

![3D topology of 27 nodes](image)

Figure 3.9: 3D topology of 27 nodes
In networks having a three dimensional layout of nodes, heat dissipation of the nodes is much more challenging than that in a two dimensional layout. So prevention of formation of hotspots having high temperatures is extremely necessary for such topologies. However in in-vivo biomedical sensor networks, a three dimensional layout of sensor nodes often becomes imperative for certain applications. For example the sensor nodes deployed to monitor an internal organ.

The performance of HPR has been compared with that of SHR and TARA for a three dimensional topology of 27 nodes (3x3x3), as shown in Figure 3.9. As in the mesh-torus topology, for comparing the protocols in an unbiased manner the value of the threshold used by TARA is set (THRESH_TARA=7.00), such that both TARA and HPR produce similar packet delivery delays at a moderately high packet arrival rate (2.5 packets per simulation interval). The results are shown in Figures 3.10, 3.11 and 3.12.

![Figure 3.10: Maximum temperature rise vs. Packet arrival rate using THRESH_TARA=7.00 in 3D topology of 27 nodes](image)
Figure 3.11: Average packet delivery delay vs. Packet arrival rate using THRESH_TARA=7.00 in 3D topology of 27 nodes

Figure 3.12: Number of packets dropped vs. Packet arrival rate using THRESH_TARA=7.00 in 3D topology of 27 nodes
Figs. 3.10, 3.11 and 3.12 show that in the 3D topology too the HPR algorithm outperforms the SHR and TARA algorithms. HPR produces much lower maximum temperature rise of the sensor nodes as compared to SHR and TARA and thus preventing the formation of hotspots having extremely high temperatures. Thus it is well suited for biomedical applications of in-vivo sensor networks those require a 3D deployment of sensors. Use of TARA leads to very large packet delivery delay and significant number of packet drop when the packet injection rate to the network becomes high.

3.4.3. Topology consisting of 100 densely connected nodes

![Figure 3.13: Network topology of 100 densely connected nodes](image)

To verify the scalability of the HPR algorithm simulation was also run over a sensor network topology consisting of 100 randomly connected nodes as shown in Figure 3.13. In the 100 nodes topology, the value of the threshold used by TARA is set (THRESH_TARA=11.00), such that both TARA and HPR produce almost similar packet delivery delays at a moderately high packet arrival rate (2.5 packets per simulation interval). The results obtained are shown in Figures 3.14, 3.15 and 3.16.
Figure 3.14: Maximum temperature rise vs. Packet arrival rate using THRESH_TARA=11.00 in topology consisting of 100 randomly connected nodes.

Figure 3.15: Average packet delivery delay vs. Packet arrival rate using THRESH_TARA=11.00 in topology consisting of 100 randomly connected nodes.
Figure 3.16: Number of packets dropped vs. Packet arrival rate using THRESH_TARA=11.00 in topology consisting of 100 randomly connected nodes

The results obtained in the topology of 100 nodes are similar to those obtained in the mesh-torus, and 3D topologies and the HPR algorithm clearly performs the best. Even at high packet injection rates the HPR algorithm produces low maximum temperature rise of the nodes and packet delivery delays. The number of packets dropped is also negligible.

Although in this topology TARA performs better than it does in mesh-torus and 3D topologies in terms of reducing the packet delivery delay, it drops a large number of packets when the packet arrival rate is high. The maximum temperature rise of the nodes when using TARA is also significantly higher than that when HPR is used.

This shows the scalability of the HPR algorithm and supports the fact that HPR can also be used in large biomedical sensor networks of the future.
3.4.4. Lifetime of a network

The lifetime of the sensor network is a very important factor when evaluating the design of any protocol for sensor networks. Although in most biomedical sensor network applications the in-vivo nodes can be recharged using infrared radiation, frequent recharging of the nodes is not desired. So the routing protocol should aim at increasing the lifetime of the sensor network. The network of embedded sensor nodes should be able to operate properly for a reasonable length of time without recharging the nodes.

A sensor network in general does not become useless if a few nodes die out. The network might still continue to route data packets to the base-station. For example if in a network of sensor nodes a few peripheral nodes die out, the network does not get disconnected and continues to operate. Even if the network gets disconnected and it is comprised of only one major component and other extremely small components, the network still remains functional as most of the nodes (in the large component) can route the packets to the base-station, provided the base-station is in the large component. However if the number of components become too many or most of the components are considerably big, a large number of nodes fail to route the data to the base-station. So the network loses its operating ability and we call the network dead or non-functional.

The lifetime of the networks using the three different routing protocols has been compared. Lifetime has been defined in the following ways:

A. Definition #1: A network is considered alive as long as 40% of the nodes are alive.

B. Definition #2: A network is considered alive as long as all the alive nodes remain connected.
We define a network as functional or operational as long as it has more than 40% of the total number of nodes alive, less than 5 components and all the components except one are of size less than 10% of the total number of nodes in the network.

![Lifetime of Network](image)

**Figure 3.17: Lifetime of the network using the three routing algorithms in three different topologies (When at least 40% of the nodes are alive)**

It can be seen from Figure 3.17 that the network has longest lifetime for all topologies when using the SHR algorithm (when Definition #1 defines lifetime). The network lifetime is the minimum when using TARA. This is due to the fact that the path chosen by the SHR for routing is the shortest and thus it drains the power of only a few nodes. TARA on the other hand withdraws packets from regions of high temperature and reroutes them using arbitrarily long paths and thus draining the power of a large number of nodes, resulting in reduced network lifetime. Using HPR algorithm results in much longer network lifetime than when using TARA as networks using HPR always try to
route packets by bypassing the zones of high temperature that appear in the path and at the same time avoids choosing arbitrarily long paths to route packets.

Figure 3.18: Lifetime of the network using the three routing algorithms in three different topologies (When all the alive nodes remain connected)

Similar reasoning holds for the results shown in Figure 3.18, from which it can be seen that as far as maintaining the connectivity of the network is concerned, the performance of HPR is comparable with that of SHR. The use TARA results in the network getting disconnected in a short time.
Figure 3.19 shows that when HPR algorithm is used the network remains functional or operational for significantly longer duration than it does when using TARA.

3.5. Conclusion

This paper presents a routing algorithm (HPR) that can be used for in-vivo biomedical sensor network. The algorithm helps reduce the rise in temperature of the nodes in the network and hence prevents the formation of hotspots having extremely high temperature. This reduces the possibility of tissue damage due to overheating of embedded sensor nodes. Although TARA performs better then SHR in terms of reducing the maximum temperature rise of nodes, HPR clearly performs much better than TARA.

At moderate and high network loads the use of TARA results in the packets incurring large average packet delivery delays. The HPR algorithm results in the packets having much lower average packet delivery delay as compared to the same when TARA
is used. Hence HPR is also suitable for being used in delay sensitive and real-time biomedical applications.

The use of TARA results in a large number of packets being dropped, even at moderate network load, whereas the use of both HPR and SHR results in negligible packet drops.

Unlike TARA, HPR also helps the embedded network to have a fairly long operational life. So sensor nodes using HPR algorithm will need to be recharged much less frequently and will operate reliably for a longer time than those using TARA, as is desired. Thus HPR routing algorithm paves the way towards development of new applications of wireless sensor network in the field of medical science.
4.1. Introduction

Although the wireless sensor motes have very limited battery power and communication and computation capabilities, the range of application of wireless motes is vast. The innumerable applications of wireless motes include intrusion detection, security and tactical surveillance, disaster management, weather monitoring, habitat monitoring applications and monitoring of different biological parameters inside the human body [6], [47-48].

In many applications these cheap motes need to be deployed in thousands as they have very limited transmission and sensing ranges. Also use of greater number of sensors results in sensing ability over larger areas with greater accuracy [47-48]. One of the problems of large-scale mass production of cheap sensor motes and their large-scale deployment is that in near future it will not be possible to assign unique hardware identifiers to each mote manufactured [48-50]. Conventional communication protocols assume that the motes are uniquely identifiable by the hardware identifiers. So such protocols will not work in a network of id-less sensor motes.

One of the most novel applications of wireless sensor networks in recent years has been in the fields of bio-medical research. Tiny sensor motes can be implanted inside human body and its different organs for monitoring and studying different biometrics and hence help in medical diagnosis [1-3], [6-7]. Thus these tiny sensor motes will bring a very radical change to medical science and will have lots of benefits to individuals and the society as a whole [1-3], [6], [11].
Although the thermal aware routing algorithms discussed the previous chapters are suitable for being used biomedical sensor networks, all of them assume that the nodes in the network are uniquely identifiable by their unique hardware ids. So these algorithms will fail to perform the desired operations when used in a biomedical sensor network of id-less sensor motes.

The authors in [49-50] have proposed an architecture that will work in a network of id-less sensors. However such a system will not be suitable for being used in a biomedical application of in-vivo sensor motes mainly due to the fact that the authors have assumed the existence of a highly powerful training agent, having long range radio, directional transmission capabilities, capability of having differential transmission power and a steady source of power. Having such a training agent is impossible in an in-vivo network.

Thus most conventional sensor network architectures will fail when the nodes in the sensor network lack unique hardware ids. The only conventional routing algorithms that will work in such networks are broadcast based algorithms like flooding. However flooding results in a large number of redundant communications and is not very suitable for energy constrained sensor networks.

In this chapter we propose the Routing Algorithm for network of homogeneous and Id-less biomedical sensor Nodes (RAIN), that will route data from the sensors efficiently to the sink in a network of id-less biomedical sensor nodes. The nodes running RAIN route data in an intelligent way so as to prevent the formation of zones having very high temperature in the network and at the same time reduce the average energy
consumption of the nodes. At the same time RAIN also achieves very high percentage packet delivery and low average packet delivery delay.

4.2. Proposed Algorithm

We assume that the in-vivo network is being used for a data gathering application, in which data packets generated by the sensor nodes need to be routed to a fixed SINK or Base-station through the multi-hop network. The Routing Algorithm for network of homogeneous and Id-less biomedical sensor Nodes (RAIN) achieves the goal of efficient communication in an in-vivo network of biomedical sensors by using a novel algorithm. The main idea of RAIN is that instead of trying to achieve the impossible task of global coordination among the sensor nodes, it focuses on local coordination.

A node running RAIN is only concerned with routing the generated and incoming data packets efficiently to its neighbors while conforming to the constraints of in-vivo sensor networks. To solve the problem of data routing in network of id-less nodes, RAIN assumes that each node has a 16-bit random number generator. Each node generates a random number between 1 and $(2^{16}-1)$, which will serve as its node-id during its operational lifetime of the node. This node-id is local and used and known by neighbor nodes only. In typical sensor network topologies the number of neighbors of a node $(N)$ does not usually exceed 10. So the probability of an id-collision in a local neighborhood is given by:

$$Pc = 1 - \left( \frac{(2^{16} - 1)(2^{16} - 2)(2^{16} - 3) \ldots \ldots (2^{16} - 11)}{(2^{16} - 1)^{11}} \right)$$
\[
= 1 - \left( \prod_{i=0}^{N} \frac{(2^{16} - 1) - i}{(2^{16} - 1)^{N+1}} \right)
\]

So the probability of id collisions is extremely small and can be safely neglected.

Even such rare occurrences of id collisions can be prevented if in the setup phase the nodes backoff for random intervals after generating the random ids, before sending the “Hello” packets to the neighbors. The neighbors after receiving the “Hello” packet, send “HelloACK” packets to respective neighbors, containing the id from the received “Hello” packet. If a node is in its backoff phase and overhears a “HelloACK” packet containing the same id that it has generated, it regenerates its id and thus avoids id-collision. The same node-id however might exist without any problem in two different localities in a network. The algorithm is designed in such a way that such duplicate network ids do not affect the performance of the network. The network scalability is also not affected. Each node conveys its id to the neighbors by broadcasting “Hello” packets containing its id. The id “0” is reserved for the SINK. The SINK sets its id to “0”. So the neighbors of the SINK know about its presence in their locality by seeing “id=0” in the received “Hello” packet.

In order to prevent the “Energy-hole problem”, in which nodes around the sink become depleted of energy very fast, the SINK broadcasts a “Status Update” message to its neighbors on receiving a packet. The “Status Update” message contains the unique packet-id of the received packet and prevents transmission of multiple copies of the same packet to the SINK by its neighbors and thus saving energy of the nodes around the SINK, preventing the “Energy-hole problem”.

66
The Routing Algorithm for network of homogeneous and Id-less biomedical sensor Nodes (RAIN) operates in three main phases, the setup phase, the routing phase and the status update phase. RAIN is fault-tolerant and the network continues to function efficiently, even when some of the nodes die out due to depletion of energy.

- **Setup Phase:**
  - The basic assumption of the RAIN algorithm is that each node has a random-number generator.
  - Each node generates a random number, which will serve as its node-id during the operational lifetime of the node.
  - The SINK sets its id to “0”.
  - Each node, including the SINK conveys its id to the neighbors by “Hello” packets containing its id.

- **Routing Phase:**
  - The packets generated by the nodes due to sensing event need to be routed to the SINK through the multi-hop network. At each hop, the hop-count of the packet is incremented by 1. Each packet generated by a node will have an unique packet-id, having the format $[N, T, R]$, where $N$ is the node-id of the node that generated the packet, $T$ is the time at which the packet was generated and $R$ is a random number. This will make the probability of packet-id collisions negligible.
  - Each node maintains a FIFO queue (id-queue) of length $L$ containing the packet-ids of the packets, having $[\text{hop-count} > \text{HOP\_THRESH}]$, that it has seen in the recent past. The \text{HOP\_THRESH} is set by experiments.
The value of \( TTL \) (Time To Live) for the data packets depends on the diameter of the network.

Each node maintains an estimate of the temperature of the neighbors. The Medium Access Control (MAC) layer of the node overhears frame transmissions of the neighbors and counts the number of frames transmitted by each neighbor in a fixed time interval. It passes the information to the Network layer periodically, so that the Network layer can estimate the temperature of the neighbors.

When a node gets a packet to route, it does the following:

- It checks the hop-count of the packet.
- If \( \text{hop-count} > TTL \)
  - The packet is dropped.
- Else If (packet-id is in id-queue)
  - The packet is dropped.
- Else If \( \text{hop-count} > \text{HOP\_THRESH} \) && packet-id is not in id-queue
  - packet-id is added to the id-queue.
- If (Packet has not been dropped)
  - If (node has the SINK in its list of neighbors)
    - The packet is delivered to the SINK.
  - Else If (node does not have the SINK in its list of neighbors)
    - The packet is forwarded to the \( n^{th} \) neighbor, other than the sender of the packet with probability \( p_n \). \( p_n \) is calculated by a function \( f(t_n) \) such that \( p_n \) is inversely proportional to the estimated temperature of the neighbor \( (t_n) \). \( T_L \) is the average
estimated temperature of the local nodes. $K$ is a constant, set by experiments.

\[
p_n = \frac{1}{\left(\frac{t_n \cdot K}{T_L}\right) + 1}
\]

- If (Packet has not been routed to any neighbor)
  - Packet is sent to “coolest” neighbor or the neighbor which has the least estimated temperature.

- **Status Update:**
  - This is necessary to prevent duplicate packet receptions at the SINK and conserve the energy of the nodes around the sink, thus reducing the energy-hole problem.
  - Whenever the SINK receives a packet, it broadcasts an “Update” message, containing the packet-id of the received packet to its neighbors.
  - Any node on receiving an “Update” message, adds the “packet-id” conveyed in the message to the id-queue.

- **Fault Tolerance:**
  - When a node has only 2 percent of its battery power left, it sends a “Bye” message to its neighbors.
  - A node on receiving a “Bye” message removes the sender of the message from its list of neighbors. This is done to prevent the nodes from choosing routes going through dead nodes.
4.3. Simulation Results

The simulation program has been developed in C++. We performed the simulation over two different sensor network topologies and compared the performance of RAIN with that of C-FLOOD. We also compared the operational life of the network when using the two different algorithms in two different topologies. C-FLOOD or Controlled-FLOOD is an enhanced version of conventional flooding algorithm. The modifications to flooding have been made for a fair comparison and adapt flooding to work in a network of id-less nodes. As in RAIN, C-FLOOD also uses local node-ids. Each node running C-FLOOD algorithm maintains an id-queue of length $L$, where it keeps the packet-ids of all the packets routed by the node in the recent past. When a node gets a packet to route, it checks the id-queue for the packet-id. If the packet-id is in the id-queue or hop-count $> \text{TTL}$, the node drops the packet. Else the node sends the packet to all its neighbors, except the one from which it received the packet. C-FLOOD also uses the “Status Update” message to prevent energy holes.

The metrics used for comparison of the protocols are:

A. Maximum temperature rise of the nodes in the network: The maximum temperature rise of the nodes in the network is a very important metric for evaluation of any algorithm used for in-vivo applications of biomedical sensor networks as a high temperature of the in-vivo nodes can damage the surrounding tissue.

B. *Average energy consumption of the nodes in the network*: Average energy consumption of nodes in the network is a very important metric for evaluating the performance of any algorithm used in sensor networks. A low average
power consumption of the wireless sensor nodes results in longer operational life of the deployed network.

C. **Percentage packet delivery:** This gives the measure of number of packets delivered to the sink as a percentage of the total number of packets generated by the nodes in the network. A good routing algorithm should result in a very high percentage packet delivery.

D. **Average packet delivery delay:** Average packet delivery delay is measured by the average number of hops taken by packets to reach the SINK. Although not always a very critical performance metric, the average packet delivery delay is important for delay sensitive and real-time applications of biomedical sensor networks.

In the simulation each second was represented by 10 simulation intervals. Each simulation run lasted 36,000 simulation intervals and thus simulated 1 hour of network operation. The performances of the two algorithms were compared at different packet injection rates. Packet injection rate in the network was varied from 0.25 to 2.00 packets per simulation interval. The sources of the packets were randomly generated and packet injection or packet arrival followed a Poisson distribution.

Studies have shown that in sensor nodes the power expended in transmission of a data packet is roughly double that expended in reception of the same. In our simulation each packet transmission by a node consumes 0.2 units of energy and each packet reception consumes 0.1 units of energy. Similarly each packet transmission by a node results in an increase in its temperature by 0.02 units and each packet reception results in increase in temperature by 0.01 units. The heat generated by the nodes due to
communication among themselves gets dissipated with time, leading to a drop in the temperature of the nodes. The cooling rate of the system (for all the nodes) has been assumed to be 0.02 units per simulation interval. The rationale behind assuming this cooling rate is as follows. Since each transmission of a packet by a node increases its temperature by 0.02 units, the cooling rate of the network of in-vivo nodes should be at least 0.02 units per simulation interval for its stable operation. The length $L$ of the id-queue for both RAIN and C-FLOOD is set to 20. The length of the id-queue is kept small since sensor nodes are usually highly energy-constrained.

The simulation results are given with 95% confidence intervals using the batch-means method with 50 batches (confidence intervals are shown as vertical bars on the plotted graphs, wherever applicable).

4.3.1. 11x11 regular mesh topology

![11x11 mesh topology with SINK at the center](image)

Figure 4.1: 11x11 mesh topology with SINK at the center

One of the most commonly used network topologies is the regular mesh. In the 11x11 regular mesh topology shown in Figure 4.1, the pairs of neighbor nodes are
connected by links. In the simulation for this topology the value of Time To Live (TTL) is set to 60 hops for both RAIN and C-FLOOD. Based on experiments the constant $K$ of RAIN has been set to 3.25 and the constant $HOP_{THRESH}$ to 20.

![Maximum Temperature Rise of Any Node vs. Load](image.png)

**Figure 4.2: Maximum temperature rise of nodes vs. Network load**

Figure 4.2 shows that with increasing packet injection rate in the network, the maximum temperature rise of the nodes in the network increases when using RAIN or C-FLOOD. However the maximum temperature rise of the nodes in the network is much higher when using C-FLOOD than when RAIN is used. The network using C-FLOOD experiences a very steep rise in maximum temperature rise with increase in packet injection rate, which makes it unsuitable for being used in in-vivo sensor networks. The maximum temperature rise in the network using RAIN rises slowly with increase in packet injection rate. So RAIN is suitable for being used in in-vivo networks.
From Figure 4.3 it can be seen that the average energy consumption per node with increasing packet injection rate has similar characteristics to the maximum temperature rise of the nodes in the network. RAIN clearly performs much better than C-FLOOD, since a network using it has much lower energy consumption per node, as compared the energy consumption when C-FLOOD is used. So RAIN is much more suitable than C-FLOOD for being used in energy constrained in-vivo sensor networks.

![Average Energy Consumption Per Node vs. Load](image)

Figure 4.3: Average energy consumption per node vs. Network load

Figure 4.4 shows that although RAIN performs much better than C-FLOOD in terms of reducing the maximum temperature rise and average energy consumption of the nodes in the network, it does not suffer much in terms of the percentage packet delivery achieved. The percentage packet delivery when using RAIN is only slightly less than that obtained using C-FLOOD.
Figure 4.4: Percentage packet delivery vs. Network load

Figure 4.5 shows that the average packet delivery delay in the network using RAIN is slightly higher than that when using C-FLOOD. Thus it can be seen that although RAIN tries to avoid routing packets through zones in the network having high temperature, it does not make the packets take unnecessarily long paths in an attempt to bypass such zones of high temperature. This shows the effectiveness of RAIN in performing desired goals.
Figures 4.6 and 4.7 show the energy consumption of the entire network when using RAIN and C-FLOOD respectively at a high network load (Packet injection rate=2.0 packets per simulation interval). For all the nodes of the network, energy consumptions are clearly much less when using RAIN than when using C-FLOOD. In both RAIN and C-FLOOD the SINK uses “Status Update” message as discussed earlier. The effectiveness of this technique in reducing the energy hole problem can be seen from Figures 4.6 and 4.7, which clearly show that the SINK and its neighbors consume much less energy than the other nodes in the network.
Figure 4.6: Energy consumption of all the nodes in the 11x11 mesh topology using RAIN at high load
Figures 4.8 and 4.9 show that even at low network load (Packet injection rate=0.25 packets per simulation interval), C-FLOOD results in much higher energy consumption of the nodes as compared to that when RAIN is used. So even at low network load RAIN performs significantly better than C-FLOOD in conserving the energy of the nodes.

Thus the simulation results clearly show that RAIN performs much better than C-FLOOD, as far as conserving the nodes’ energy is concerned under all network load conditions.
Figure 4.8: Energy consumption of all the nodes in the 11x11 mesh topology using RAIN at low load
Figure 4.9: Energy consumption of all the nodes in the 11x11 mesh topology using C-FLOOD at low load
Figure 4.10: Maximum temperature rise of all the nodes in the 11x11 mesh topology at high load. The upper mesh is for C-FLOOD and the lower one is for RAIN

The efficiency of RAIN in reducing the maximum temperature rise of all the nodes in the network at both high and low loads can be seen in Figures 4.10 and 4.11. The use of “Status Update” message by the SINK results in the SINK and its neighbors having a low rise in temperature. RAIN clearly performs much better than C-FLOOD in reducing the maximum temperature rise of the nodes in the network.
Figure 4.11: Maximum temperature rise of all the nodes in the 11x11 mesh topology at low load. The upper mesh is for C-FLOOD and the lower one is for RAIN.

4.3.2. Topology consisting of 100 randomly connected nodes

Figure 4.12: Topology consisting of 100 randomly connected nodes with SINK in the middle
We compared the performance of RAIN and C-FLOOD over a randomly connected irregular topology shown in Figure 4.12 since sensors are sometimes deployed randomly because of the large number of sensors deployed or other application specific requirements. In the simulation for this topology the value of Time To Live (TTL) is set to 50 hops for both RAIN and C-FLOOD. Based on experiments the constant $K$ of RAIN has been set to 4.50, and the constant $HOP\_THRESH$ to 25.

MAXIMUM TEMPERATURE RISE OF ANY NODE VS. LOAD

![Graph showing maximum temperature rise of nodes vs. load](image)

Figure 4.13: Maximum temperature rise of nodes vs. Network load

From Figure 4.13 it can be seen that as in the regular mesh topology, in the randomly connected topology too RAIN performs much better than C-FLOOD in terms of reducing the maximum temperature rise of the nodes. Use of C-FLOOD results in even higher temperature rise of the nodes than it did in the regular mesh topology.
Figure 4.14 shows that the average energy consumption of the nodes in the randomly connected topology has similar characteristics to the same in the regular topology. Use of RAIN results in much lower average energy consumption of the nodes than when C-FLOOD is used.

**AVERAGE ENERGY CONSUMPTION PER NODE VS. LOAD**

![Graph showing average energy consumption per node vs. network load. The graph compares RAIN and C-FLOOD methods. RAIN has a lower average energy consumption across all load levels.](image)

**Figure 4.14: Average energy consumption per node vs. Network load**
As seen in Figure 4.15, use of RAIN in the randomly connected topology results in nearly 100% packet delivery. The percentage packet delivery in this topology is even higher than that obtained the regular mesh topology due to the presence of larger number of alternative paths to route packets.
Figure 4.16 shows that the average packet delivery delay when using RAIN is slightly higher than that when C-FLOOD is used.

The simulation results show that although use of C-FLOOD results in slightly higher percentage packet delivery and slightly lower average packet delivery delay than when RAIN is used, the network using C-FLOOD has very high temperatures and energy consumptions in majority of the nodes. If such a network is allowed to operate for a considerable amount of time, it will definitely result in significant damage of the tissues surrounding the embedded nodes. The reason for the poor performance of C-FLOOD is that the nodes using C-FLOOD do not consider the temperature rise of the nodes when routing packets and continues forwarding packets to neighbors having high temperature and thus raising their temperature even higher. Also in an attempt to achieve 100%
packet delivery, a node using C-FLOOD routes packets to all its neighbors except the sender and thus having a large number of redundant packet transmissions and receptions and increased average energy consumption of the nodes.

Nodes in the network using RAIN on the other hand keeps an estimate of the temperature of the neighbor nodes and forwards packets to relatively more heated nodes of a locality with extremely small probability. So the heated nodes get time to cool down by dissipating heat and thus the nodes in the network never have unacceptably high temperatures. The probabilistic packet forwarding also reduces redundant packet transmissions and thus reducing the average energy consumption of the nodes. The slight increase in average packet delivery delay when using RAIN is due to the fact that RAIN routes packets by bypassing heated zones in the network and thus increasing the packet delivery delay slightly. A node using RAIN forwards a packet to the coolest neighbor if it fails to probabilistically forward it to any neighbor. This reduces the chances of packets being dropped due to formation of zones of relatively higher temperature in the network and thus helps achieve very high percentage packet delivery. RAIN also puts a packet-id in id-queue only when the hop-count for that packet is greater than a certain value $HOP-THRESH$. This allows a bounded redundancy in packet transmissions in the network that helps achieve a high percentage packet delivery and at the same time prevent routing loops in the network.
4.3.3. **Operational life of the network**

The operational life of the sensor network is a very important factor that needs to be considered when evaluating the design of any sensor network protocol. Although in most biomedical sensor network applications the in-vivo nodes can be recharged using infrared radiation, frequent recharging of the nodes is not desired. So the routing protocol should aim at increasing the operational life of the sensor network. The network of embedded sensor nodes should be able to operate properly for a reasonable length of time without recharging the nodes.

To compare the operational life of the networks we have used two measures:

A. Minimum length of time the network operates before there is any dead node in the network.

B. Minimum length of time the network operates with more than 50 percent of the nodes being able to reach the SINK. A node might not be able to reach the SINK if it is dead or there is no path from the node to the SINK due to death of nodes in the prospective paths.

Here we have assumed that node-death occurs only due to energy depletion, i.e. when the batteries in the nodes drain out fully. Also to obtain the results fast, each of the nodes are assumed to have an initial energy of 100 units only.
Figures 4.17 and 4.18 show the death of the nodes in the network having 11x11 mesh topology when using RAIN and C-FLOOD under low and high network load respectively. At both low and high network loads most nodes die much earlier when using C-FLOOD than when RAIN is used. However at high load some nodes near the SINK die after a long time when using C-FLOOD. This can be attributed to the fact that since most nodes away from the SINK die very early when using C-FLOOD; the nodes near the SINK do not get much data to route and thus live long, although the network may no longer be useful. It can also be observed that under both high and low network loads the number of node deaths is much more when using C-FLOOD than when RAIN is used.
Figure 4.18: Death of the nodes in the 11x11 mesh topology when using RAIN and C-FLOOD under high network load
Figure 4.19: Operational life of the network before any node dies when using RAIN and C-FLOOD in different network topologies under low load.

Figure 4.19 shows that in both topologies use of C-FLOOD at low load results in the network having a node dead much earlier than when RAIN is used.

As shown in Figure 4.20, C-FLOOD performs even worse when the network load is high.
Figure 4.20: Operational life of the network before any node dies when using RAIN and C-FLOOD in different network topologies under high load.
Figures 4.21 and 4.22 show that the network becomes unusable (more than 50 percent of the nodes are unable to reach the SINK) much more rapidly when using C-FLOOD than when RAIN is used.

Thus the simulation results clearly show that use of RAIN gives the network a much longer operational life than when using C-FLOOD.
Figure 4.22: Operational life of network with more than 50 percent of the nodes being able to reach the SINK when using RAIN and C-FLOOD in different network topologies under high load

4.4. Conclusion

This chapter presents the Routing Algorithm for network of homogeneous and Id-less biomedical sensor Nodes (RAIN) that can be used in in-vivo biomedical sensor networks of id-less sensor nodes. Use of RAIN results in low average energy consumption of the sensor nodes and low temperature rise of the nodes and thus conforming to the constraints of in-vivo networks. At the same time use of RAIN also achieves very high percentage packet delivery and low average packet delivery delay. RAIN also gives the network a long operational life. Simulation results show that RAIN performs much better than C-FLOOD, a controlled flooding algorithm when used in a biomedical sensor network.
CHAPTER 5
BIOCOMM – CROSS-LAYER MAC AND ROUTING PROTOCOL
CO-DESIGN FOR BIOMEDICAL SENSOR NETWORKS.

5.1. Introduction

In Biomedical Sensor Networks energy efficient communication is needed while keeping in mind the additional constraints imposed by such in-vivo networks. In this research we are studying the effect of the MAC layer and Network layer on energy efficient communication in such networks. Although the thermal aware routing algorithms proposed in Chapters 2, 3 and 4 are suitable for being used biomedical sensor networks, the nodes running these algorithms need to estimate the temperature of the neighbors by overhearing their transmissions. So the Medium Access Control (MAC) protocol used by them is an “always on” MAC in which the node’s radio always remains “On” and listens to the wireless channel. Use of such “always on” MAC protocols like IEEE 802.11 [19, 20] results in significant power drain of the nodes due to overhearing and idle-listening and are not suitable for being used in highly energy-constrained wireless sensor networks [21]. So communication protocols need to be developed for use in Biomedical Sensor Networks such that they prevent the formation of hotspots in the network and at the same time use the nodes’ energy efficiently.

Till date researchers have tried to prevent formation on hotspots in Biomedical Sensor Networks by designing suitable Routing algorithms. The role of the MAC layer protocol to prevent formation of hotspots and optimize the overall performance of in-vivo networks remains largely unaddressed.

The Medium Access Control (MAC) protocol allows wireless nodes to efficiently share the common wireless medium [21]. Different scheduled and contention based
Medium Access Control protocols have been proposed for Wireless Sensor Networks [21, 22]. However none of the protocols developed addresses the issues specific to in-vivo biomedical sensor networks.

The MAC protocol used can have a significant impact on the performance of the routing protocol used in any network [20, 23]. Contention based MAC protocols are more suitable for being used in in-vivo networks as compared to scheduled protocols because of their lower overhead.

In this Chapter we propose Biocomm, which is a cross-layer Medium Access Control (MAC) and Routing protocol co-design for Biomedical Sensor Networks. The need for cross-layer protocol designs has been emphasized by the authors in [23, 24]. It might not be sufficient to optimize the performance of the network or the MAC layers in isolation. In Biocomm the Network and the MAC layers interact with each other to optimize the overall performance of the network of in-vivo wireless biomedical sensor motes. The MAC protocol used in Biocomm gives less transmission priority to nodes having higher temperature in an attempt to prevent formation of hotspots. It also gives prioritized medium access to nodes experiencing buffer congestion in order to reduce network congestion and packet drops due to buffer overflow.

5.2. Proposed Architecture

The proposed cross-layer architecture for Biocomm protocol has been shown in Figure 5.1. In the proposed Biocomm protocol the Network and the Medium Access Control (MAC) layers interact with each other to optimize the overall network performance. The interaction of the two layers is facilitated by the Cross-layer
Messaging Interface (CMI). With the help of CMI, the network layer sends its status information to the MAC layer and vice-versa.

The network layer keeps track of the vacant space remaining in its packet buffer (BS). This information is sent to the MAC layer through the CMI, which allows the MAC layer to give higher frame transmission priority to the congested nodes, in an attempt to reduce packet drops in the network layer due to buffer overflow. The MAC layer is assumed to have a very small frame buffer. The frame buffer does not overflow as the
MAC layer accepts packets from the Network layer only when its buffer has empty space.

The MAC layer maintains a *Neighbor Status Table*. The status of the neighbors is set to *Free (F)* or *Blocked (B)* by the MAC logic. The Network layer also maintains a *Neighbor Status Table*. Whenever the status of any neighbor in the *Neighbor Status Table* changes, a “*BLOCK*” or “*FREE*” message is generated and sent to the Network layer with the help of *CMI*. The Network layer on receiving the message updates the status of the associated neighbor in its *Neighbor Status Table*.

The MAC layer also keeps track of the most recent active node (*LA*) by overhearing RTS and CTS packets. This information is also sent to the Network layer with the help of *CMI*. The neighbor status information and the Last active node (*LA*) information help the network layer avoid routing packets through paths having sleeping nodes.

Biocomm uses a RTS/CTS based scheme, similar to the one used in IEEE 802.11. The RTS and CTS frame formats used are also same as those used in IEEE 802.11 (Shown in Figures 5.2 and 5.3). In the frame formats TA and RA denote Transmitter Address and Receiver Address respectively.

![Figure 5.2: Request To Send (RTS) Frame Format (Standard 802.11)](image)
5.3. Proposed Protocol

5.3.1. Medium Access Control (MAC)

Frame Transmission:

\[
\text{Backoff} = \text{rand}(0,1) \times W + C1 \times \text{rand}(0,1) \times \left( \frac{\delta}{B} \right) + C2 \times \text{rand}(0,1) \times (t - T1)
\]

- \( \text{rand}(0,1) \) generates random number between 0 and 1.
- \( W = \text{Min}(2^{2^i}, 32) \). \( i \) = Number of RTS transmission attempts.
- \( \delta \) is the remaining buffer space in the Network layer. This information \( \text{(BS)} \) is transferred to the MAC layer from the Network layer with the help of \( \text{CMI} \).
- \( B \) is the total capacity of the packet buffer in the Network layer.
- \( t \) is the instantaneous temperature of the node.
- \( T1 \) is a temperature threshold beyond which a node gets reduced transmission priority.
- \( C1 \) and \( C2 \) are constants.

- If \( (t \leq T1) \)
  - Generate small backoffs.
  - \( C1 = \) Positive constant \( > 0 \) [Set by experiments].
  - \( C2 = 0. \)
- If \( t > T1 \text{ and } t \leq T2 \) [\( T2 > T1 \)]
  - Node has less transmission priority – Generate large backoffs.
  - \( C1, C2 = \text{Positive constants} > 0 \text{ [Set by experiments]} \).

- If \( t > T2 \)
  - Turn off radio for estimated time \( D \). \( D = \text{Time needed to cool down node from } T2 \text{ to } T1 \). After time \( D \), the node turns on the radio and sends a “Dummy RTS frame” with \( RA=\text{Node-id}, TA=\text{Node-id} \text{ and } Duration=0 \).

- If (RTS has been sent for a frame \( RTS\_THRESH \text{ times, and still no CTS is received} \)
  - Drop the frame.
  - Mark the status of node \( M \) as “Blocked” (B) in the Neighbor Status Table, where \( M=RA \) of the RTSs sent.
  - \( CMI \) sends a message to Network layer: “BLOCK M”.
  - The Network layer on receiving “BLOCK M”, marks neighbor \( M \) as “Blocked” (B) in its Neighbor Status Table.

**Overhearing:**

- If the status of the node having id \( M = TA \) of the overheard RTS or the \( RA \) of a overheard CTS is “Blocked” (B):
  - Set status of node \( M \) as “Free” (F) in the Neighbor Status Table.
  - \( CMI \) sends a message: “FREE M” to Network layer.
  - The Network layer on receiving “FREE M” marks \( M \) as “Free” (F).

- MAC layer notifies the network layer about the \( TA \) of the last RTS and the \( RA \) of the last CTS that it heard through the \( CMI \) and also keeps track of the same in a variable \( LA \).
- Network layer stores the received address \((RA/TA)\) in a variable \(LA\).
- Energy Saving: On hearing CTS for which \(RA \neq \text{Node-id}\), a node turns off its radio and sleeps for time \(S = \text{“Duration” field of overheard CTS}\).

### 5.3.2. Routing

Each sensor node handles two types of packets, those that originate at the node itself due to sensing and those that are forwarded by other nodes, to be routed to destination. Both types of packets are treated in the same way by the node. Each node tries to route the packets in such a way that the packets get delivered to the destination using the minimum number of hops, i.e. the shortest path, unless there is a hotspot or sleeping node in the path.

- If \((hop\_count > \text{TTL (Time To Live)})\)
  - Drop packet
- Else If (status of next-hop node in the shortest path to the sink is \(\text{“Free” (F) in the Neighbor Status Table}\))
  - Increment \(hop\_count\).
  - Send packet to next-hop node.
- Else If (status of next-hop node in the shortest path to the sink is \(\text{“Blocked” (B) in the Neighbor Status Table}\))
  - Increment \(hop\_count\).
  - Send packet to node with node-id \(LA\).
5.3.3. Modifications for delay sensitive applications (Biocomm-D)

The protocol used for delay sensitive applications is similar to the one proposed for non-delay sensitive applications with certain modifications to reduce the average packet delivery delay.

Packet dropping:

Packet dropping is done in the Network layer to keep the average packet delivery low. When a node receives a packet from a neighboring node, it calculates the duration \( T \) for which the packet has circulated in the network or the age of the packet. \( \text{time()} \) gives the instantaneous global time and is the origin / generation timestamp of the received packet. The nodes need to be somewhat synchronized for this, else the number of hops taken by that packet can also be considered instead of age.

\[
T = \{\text{time()} - \tau\}
\]

\[
P_d = \frac{K \ast (T - \alpha)}{(\beta - \alpha)}
\]

The packet dropping is performed using the following algorithm:

- If \((T >= \alpha && T <= \beta)\)
  - Drop packet with probability \(P_d\)
- Else If \((T > \beta)\)
  - Drop packet
- Else If (Packet not dropped)
  - Put packet in buffer queue
In the expression for $P_d$, $\alpha$ is the age of the packet after which probabilistic dropping is performed, $\beta$ is the age of the packet after which the packet is dropped with $P_d=1.0$. $K$ is a constant set by experiments and $T$ is the age of the packet.

**Buffer Scan:**

When the temperature of a node exceeds $T_2$ and it has turned off its radio for time duration $D$, it scans its buffer for packets having ($T > \gamma$). The node removes such packets from the buffer and re-compact the buffer.

In the Biocomm protocol in the equation for backoff calculation, if a node has congested buffer ($\delta$ will have a small value), on an average the backoff generated will be smaller than that if the buffer was not congested. So this will give higher transmission priority to congested nodes and thus reducing the severity of buffer congestion. The temperature of a node also influences the backoff generated. If the temperature of a node is above threshold $T_1$, more the temperature of a node, more is the generated backoff. The above strategy reduces the rate at which the node’s temperature rises, once its temperature becomes higher than $T_1$ because transmission of a frame by a node results in much higher temperature rise of a node than reception of a frame. As discussed earlier, on receiving a packet a node enqueues it in its packet buffer if the buffer has vacant space, else it drops the packet.

### 5.4. Simulation

The simulation program has been developed in C++. We performed the simulation over a 11x11 regular mesh topology and compared the performance of Biocomm and Biocomm-D with that of Hotspot Preventing Routing (HPR) presented in
Chapter 3 and Shortest Hop Routing (SHR). In Shortest Hop Routing or SHR the packets take the shortest path from the source to the destination, disregarding the temperature rise of the nodes in the path. We have not compared the results with TARA as in Chapter 3 and [2] we have showed that HPR performs much better than TARA. In the simulations when SHR or HPR are used as the Routing protocol, the MAC protocol used is IEEE 802.11. The network topology consists of a regular layout of 11x11 nodes, as shown in Figure 5.4, with the sink at the center. In Figure 5.4 pairs of neighbor nodes are connected by links. The wireless links between neighbors is assumed to be bi-directional. The network runs a data-gathering application where the nodes need to send the sensed data to the sink through the multi-hop wireless in-vivo network. The sensor nodes have been assumed to have uniform constant sensing rate. So all the nodes in the network except the sink have same packet injection rate in the network. The network load is varied by changing the sensing rate or the packet injection rate.

The network is assumed to be homogeneous and each node can buffer 100 packets in its Network layer packet buffer. If the packet buffer becomes full the node drops the incoming packets. The size of the frame buffer in the MAC layer is assumed to be very small (Less than 5). A very large buffer in the MAC layer results in the Routing protocol having a very slow response to the network dynamics. The MAC and the Routing logic ensure that there is no frame drop in the MAC layer.

The simulation was run for 6000 simulation intervals. The nodes inject packets in the network till simulation interval 3000. Studies have shown that in sensor nodes the power expended in transmission of a data packet is roughly double that expended in reception of the same. In our simulation each packet transmission by a node consumes
0.2 units of energy and each packet reception consumes 0.1 units of energy. Similarly each packet transmission by a node results in an increase in its temperature by 0.02 units and each packet reception results in increase in temperature by 0.01 units. The heat generated by the nodes due to communication among themselves gets dissipated with time, leading to a drop in the temperature of the nodes. The cooling rate of the system (for all the nodes) has been assumed to be 0.01 units per simulation interval. The rationale behind assuming this cooling rate is as follows. Since each reception of a packet by a node increases its temperature by 0.01 units, the cooling rate of the network of in-vivo nodes should be at least 0.01 units per simulation interval for its stable operation.

For Biocomm, T1 and T2 are set to 10 and 20 respectively by experiments for optimum performance under all loads. For Biocomm-D T1 and T2 are set to 20 and 35. The value of TTL (Time To Live) is set to 60. C1=2.0, C2=0.50. For Biocomm-D $\alpha = 500$, $\beta = 5000$, $\gamma = 5000$ and K=15.0. In the simulation results the packets dropped by Biocomm-D using probabilistic packet dropping and buffer scan techniques is considered as packets dropped due to buffer congestion.

![Figure 5.4: Regular mesh topology with SINK at the center](image)
The metrics used for comparison of the protocols are:

A. *Maximum temperature rise of the nodes in the network*: The maximum temperature rise of the nodes in the network is a very important metric for evaluation of any communication algorithm used for in-vivo applications of biomedical sensor networks as a high temperature of the in-vivo nodes can damage the surrounding tissue.

B. *Average energy consumption of the nodes in the network*: Average energy consumption of nodes in the network is a very important metric for evaluating the performance of any algorithm used in sensor networks. A low average power consumption of the wireless sensor nodes results in longer operational life of the deployed network and hence increasing network reliability.

C. *Average packet delivery delay*: Average packet delivery delay is measured by the average time taken by packets to reach the sink after they are generated. Although not always a very critical performance metric, the average packet delivery delay is important for delay sensitive and real-time applications of biomedical sensor networks.

D. *Percentage of injected packets dropped due to buffer congestion*: The percentage of injected packets dropped due to buffer congestion is an important metric in determining the performance of any routing algorithm because of the large overhead incurred in retransmitting dropped packets.

E. *Percentage of injected packets dropped due to TTL expiration*: The percentage of injected packets dropped due to TTL expiration is an important metric in determining the performance of any routing algorithm because if a
large number of packets are dropped due to TTL expiration, it often means that packets are taking very long paths through the network to go to the sink.

**F. Cumulative Network Throughput:** The throughput of the network is defined as the number of data packets delivered to the sink per unit time. Throughput of a network is one of the most important performance metrics as the overall performance of any application running on the network depends on it.

The simulation results are given with 95% confidence intervals using the batch-means method with 50 batches (confidence intervals are shown as vertical bars on the plotted graphs, wherever applicable).

**Figure 5.5:** Maximum temperature rise of the nodes in the network with decreasing packet injection interval (increasing network load)
Figure 5.5 shows the maximum temperature rise of the nodes in the network with increasing network load. SHR results in highest temperature rise of any node in the network since it does not consider temperature rise of the nodes when routing data packets through the multi-hop wireless in-vivo network. Use of HPR considerably reduces the maximum temperature rise of the nodes in the network as compared to SHR.

Both Biocomm and Biocomm-D result in very low temperature rise of the nodes in the network. In an attempt to reduce the average packet delivery delay, use of Biocomm-D results in slightly higher temperature rise than when Biocomm be used. It can be observed from Figure 5.5 that maximum temperature rise of the nodes in the network rises with the increase in network load when using SHR or HPR. Both Biocomm and Biocomm-D however keep the maximum temperature rise fairly constant, even if the network load increases. This can be attributed to the fact that when using the proposed protocols the nodes turn off their radios when the temperature reaches T2. So the maximum temperature rise cannot exceed much beyond T2. So use of both Biocomm and Biocomm-D ensure that the temperature of the nodes in the network will not exceed much beyond T2.

T2 can be chosen according to application requirements. Implantation of the biomedical sensor nodes in some parts of the body like the eye might require the network to have a lower value of T2 than that required by nodes implanted in other parts of the body. In an in-vivo network all the nodes can also have different values of T2 when using Biocomm or Biocomm-D based on their region of implantation. However if the value of T2 chosen be too small, the node will turn its radio on and off too frequently and that would have a negative impact on the other performance metrics like throughput, latency
and packets dropped. In the simulation we have chosen the values of T2, such that the overall performance of the network is optimized.

Figure 5.6: Maximum temperature rise of the nodes in the 11x11 mesh topology when using SHR as the Routing protocol and 802.11 as the MAC protocol at a high network load
Figure 5.7: Maximum temperature rise of the nodes in the 11x11 mesh topology when using HPR as the Routing protocol and 802.11 as the MAC protocol at a high network load
Figure 5.8: Maximum temperature rise of the nodes in the 11x11 mesh topology when using Biocomm at a high network load
Figures 5.6, 5.7, 5.8 and 5.9 show the maximum temperature rise of the nodes in the 11x11 network topology. Figure 5.6 shows that when using SHR the network has several hotspots or regions of extremely high temperature, which makes it very unsuitable for being used in an in-vivo sensor network. Figure 5.7 shows that although use HPR reduces the temperature of the hotspots formed in the network as compared to that when SHR be used, the nodes in the layout have large variation in their maximum temperature rise. Figures 5.8 and 5.9 show that use of Biocomm or Biocomm-D results in the network having fairly uniform maximum temperature profile throughout the network.
The average energy consumption of the nodes in the network when using the different protocols can be seen from Figure 5.10. It can be clearly seen that use of both SHR and HPR result in the nodes having much higher average energy consumption as compared to that when Biocomm or Biocomm-D are used. This is due to the fact that SHR and HPR use IEEE 802.11 in the MAC layer and IEEE 802.11 is an “always on” MAC protocol, which wastes a lot of energy due to idle listening. HPR needs an “always on” MAC so that a node using HPR can estimate the temperature of the neighbor nodes by overhearing their transmissions. The nodes using HPR consumes even more energy than those using SHR as it reroutes packets preventing hotspots and thus on an average the packets take larger number of hops as compared to that when SHR be used.
Biocomm and Biocomm-D save a lot of energy by putting the nodes to sleep by turning off their radio whenever possible, as described in the protocol. The average energy consumption of the nodes when using Biocomm or Biocomm-D is reduced to almost one third that when SHR be used and one fourth that when HPR be used. Thus use of Biocomm or Biocomm-D significantly reduces the energy consumption of the nodes in the Network.

The energy savings obtained throughout the 11x11 network when using Biocomm or Biocomm-D is can be clearly seen from Figure 5.11.

Figure 5.11: Energy Consumption of the nodes in the 11x11 mesh topology
Figure 5.12: Average packet delivery delay in the network with decreasing packet injection interval (increasing network load)

Figure 5.12 shows the average packet delivery delay with increasing network load. Use of SHR results in the network having least average packet delivery delay as the packets always take the shortest path from the source to the sink. Use of HPR results in much larger average packet delivery delay as compared to that when SHR be used. Biocomm produces slightly higher delay as compared to HPR, which is not a problem because Biocomm is designed to give optimum performance for non-delay-sensitive applications. Use of Biocomm-D, which has been designed for delay sensitive applications results in much lower average packet delivery delay as compared to that when Biocomm or HPR be used. This is due to the fact that Biocomm-D drops packets
that are being circulated in the network for a very long time by probabilistic dropping and buffer scan techniques.

Figure 5.13: Percentage of injected packets dropped in the network due to buffer congestion with decreasing packet injection interval (increasing network load)

Figure 5.13 shows that the percentage packets dropped in the network rises very steeply with increasing network load when SHR is used. Although HPR has slightly higher packet drop as compared to that of SHR at low load, HPR performs better than SHR under moderate and high network loads. Use of Biocomm results in very negligible percentage of the total packets being dropped due to buffer congestion as it gives frame transmission priority to congested nodes by efficient cross-layer design. At high load the percentage of packets dropped by BOCOMM-D is less than that of SHR or HPR. The packets dropped by Biocomm-D are not due to buffer overflow, rather they are dropped
by the Probabilistic dropping and Buffer scan techniques, used to lower the average
delivery delay of the delivered packets.

It can be observed from Figure 5.14 that the percentage of total packets dropped
due to TTL expiration is very small and decreases with increase in network load when
using HPR, Biocomm or Biocomm-D. This is due to the fact that with increase in
network load a large number of packets which are being circulated in the network for a
long time get dropped due to congestion when or probabilistic dropping.

PACKETS DROPPED DUE TO TTL EXPIRATION VS. LOAD

Figure 5.14: Percentage of injected packets dropped in the network due to TTL expiration with
decreasing packet injection interval (increasing network load)
Figure 5.15 shows the cumulative network throughput with time. Use of SHR results in the highest network throughput since SHR delivers packets to the sink following the shortest path, disregarding the formation of hotspots. Another interesting observation that can be made is that after time instant 4500, the cumulative throughput for SHR starts to drop. This can be attributed to the fact that packet injection stops at time instant 3000 and the network using SHR also drops a large number of packets at high load. So by time instant 4500, very few (if any) packets are left in the network to be delivered to the sink. A network using Biocomm achieves slightly lower cumulative throughput as compared to that when HPR be used. The throughput however improves when using Biocomm-D as it uses delay-reducing techniques.
5.5. Conclusion

In this chapter Biocomm, a cross layer MAC and routing protocol co-design for Biomedical Sensor Networks has been proposed. Simulation results show that Biocomm performs much better than Shortest Hop Routing and the previously proposed Hotspot Preventing Routing (Both using IEEE 802.11 in MAC layer) in terms of reducing the maximum temperature rise of the nodes, average energy consumption of the nodes and packets dropped due to buffer congestion. A variation of Biocomm, Biocomm-D also reduces the average packet delivery, compared to HPR. Thus the research presented in this chapter takes the first step towards efficient cross-layer design of MAC and Routing protocols for Biomedical Sensor Networks.
CHAPTER 6
ECO-FRIENDLY THERMAL AWARE ROUTING PROTOCOL FOR HABITAT MONITORING

6.1. Introduction

One of the major applications of wireless sensor networks is in the field of environmental monitoring of the diverse and unique ecosystems and animal habitats. The networks of wireless sensor nodes are used to monitor the microclimates in the animal habitats and the animal behavior patterns. One of the main requirements of a habitat monitoring infrastructure is inconspicuous operation, which means the system should not alter the behavior of the species under study.

Sensor nodes, however small they are produce heat when communicating wirelessly with the neighbors. This heat results in a temperature rise of the sensor node and its surroundings. The sensors need to be placed very close to the species in order to monitor the microclimate around them. If the temperature rise of the sensor nodes is too high, it might have an adverse effect on the species, forcing a change in their behavior. So the sensor nodes deployed for studying the behavior of species in a non-invasive way should use hardware and protocols which help to minimize the amount of heat produced and hence reduce the rise in temperature of the nodes. The sensor nodes often need to measure the temperature of the microclimate being studied. A large temperature rise in the nodes may also affect the accuracy of such temperature measurements.

Existing communication protocols may not be appropriate for use in networks of sensors for habitat monitoring since such networks impose some additional constraints on the protocols used.
In this chapter, we propose a thermal aware routing protocol [4], which can be used in sensor networks used for habitat monitoring applications. The design of the protocol ensures that the amount of heat produced in the network is reduced as much as possible and none of the sensor nodes has a very high temperature rise due to communication.

To demonstrate the effectiveness of our protocol, simulations have been conducted. The simulation results show that our proposed Least Temperature Routing (LTR) protocol [4] is well suited to be used in habitat monitoring applications.

6.2. System Topology

In the sensor network that needs to be deployed we assume that there are two types of nodes:

A. Thermal aware nodes: Those nodes which are very close to the species, i.e. in their nests or burrows, run the proposed Least Temperature Routing protocol and are thermal aware.

B. Non thermal aware nodes: The set of nodes which are away from the habitat are normal sensor nodes, that need not be thermal aware and are used only for forwarding the data to the base-station, using shortest hop routing protocol.
Figure 6.1 shows an example of habitat monitoring network. The nodes inside the dotted-rectangle are the ones which are deployed inside the habitat to be monitored. A zoomed out view of this network of nodes inside the habitat region is also shown. All of these nodes are thermal aware nodes running the LTR protocol.

The circled nodes are the nodes which are thermal aware but are placed at the periphery of the habitat being monitored. These are called the peripheral nodes and they are connected with the non-thermal aware nodes placed outside the habitat for routing the sensed data to the Base Station.
The nodes inside the Habitat route the information to the peripheral nodes of the habitat using the LTR protocol. The peripheral nodes route the data to the base station using the Shortest Hop Routing (SHR) algorithm.

6.3. Proposed Protocol

The proposed Least Temperature Routing (LTR) protocol is used by the nodes inside the habitat and operates in four basic phases.

- **Setup Phase:**
  - The nodes communicate with the neighbors and gather information about their temperature.
  - The nodes discover which peripheral nodes they can directly reach, if any.

- **Routing Phase:**
  - All the thermal aware nodes inside the habitat route data packets to the peripheral nodes.
  - A node has to handle two types of packets, those that originate due to sensing events by the node itself and those that are coming from other nodes [17]. Both types of packets are treated in the same way.
  - A packet is forwarded using the LTR protocol until it reaches a peripheral node.
  - If a node having a packet to route has one peripheral node as its neighbor, the packet is forwarded directly to the peripheral node. If it is connected to more than one peripheral node, the packet is routed to the peripheral node having least temperature at that time. As in [7], we assume each node has information about the temperature of its neighbors.
If a node having a packet to route does not have any peripheral node as its neighbor, it forwards the packet to the thermal aware neighbor having the least temperature, i.e., the “coolest neighbor”.

- **Packet discard:**
  - Each packet has a hop-count which is incremented by one each time a node forwards a packet.
  - Once a node gets a packet, it checks the hop-count. If it is above a certain threshold value, MAX_HOPS, the packet is discarded.
  - The value of MAX_HOPS depends on the diameter of the sensor network.

- **Reducing unnecessary hops and loops:**
  - Each packet maintains a small list of nodes it has most recently visited.
  - If the “coolest neighbor” is already there in the list of recently visited nodes for the packet, the packet is forwarded to the neighbor having the second lowest temperature among the neighboring nodes.
  - The list of recently visited nodes should include all nodes visited within some past window. Although it is better to have large window, the size of the list is practically constrained and must be kept small in order to prevent packets from becoming too large and to reduce the overhead of searching the list. However since only one bit per node is needed in a packet to keep the history, the possibility of a packet becoming very large is not considered in this research.

The nodes that are deployed outside the habitat are used for routing the packets from the peripheral nodes of the habitat to the Base Station (BS). These nodes build the
routing table during setup and forward the packets to the base station using the shortest path.

6.4. Simulation

The simulation program was developed in C. We performed the simulation over a topology having 50 thermal aware nodes deployed in the habitat and connected in a mesh-like fashion. The topology has 16 peripheral nodes which route the gathered data using the normal sensor nodes, located outside the habitat to the base-station.

We compared the performance of the proposed protocol with the shortest hop routing protocol (SHR) with increasing packet arrival rate in the system, i.e. with increasing sensing rate of the sensor nodes. The performances of the two protocols with time were also compared.

When the nodes inside the habitat use shortest hop routing protocol, each node tries to forward the data packets to the peripheral node which can be reached with least number of hops. Since the sensor nodes perform routing functions, we shall use the terms “node” and “router” interchangeably to mean a sensor node.

The metrics used for the comparison are:

A. Average temperature rise of the nodes deployed in the habitat.

B. The maximum temperature rise of any node in the habitat.

C. The average delay in packet delivery in the network.

D. The average power consumption of the sensor nodes.

The average temperature rise of the nodes gives the average change in temperature of the nodes from that at the beginning of the simulation. We assume that
idle listening consumes one unit of power. Both transmission and reception of each packet by a node consume same amount of power units (two units) [12].

To prevent any packet from indefinitely circulating in the network, in both algorithms, it has been assumed that the maximum number of hops allowed for a packet to reach the destination is 40. If a router receives a packet that has a hop-count greater than 40, the packet is discarded.

Each simulation run generated 2000 data packets that are routed through the network. Each time a node receives a packet to route, its temperature increases by 0.1 unit. The cooling rate of the system (all the nodes) is assumed to be 0.1 unit per simulation interval.

It has been assumed that the routers have very limited buffer size, such that if a packet arrives at any router it gets routed to a neighboring router in the next simulation interval, i.e., the packet cannot be buffered any longer at the current router. Buffering of packets for a long time in the intermediate routers results in a large increase in the average packet delivery delay and should be avoided.

The average temperature rise of the nodes in routing the data packets gives an estimate of the amount of heat produced in the network by the nodes due to the wireless communication. Figure 6.2 shows the average temperature rise of the nodes deployed within the habitat with increasing packet arrival rate. It can be observed that the LTR produces lower average temperature rise than SHR for all packet arrival rates and thus performs better.

The results in Figure 6.2 can be explained as follows. SHR tries to route the packets following the shortest path, disregarding the temperature rise of the nodes in the
path and thus raising their temperature. LTR tries to route the packets through the ‘cooler’ nodes from the beginning and thus performs well even at high packet arrival rates.

![Average Temperature Rise vs. Packet Arrival Rate](image)

**Figure 6.2: Average temperature rise vs. packet arrival rate**

The maximum temperature rise of any node inside the habitat is the most important metric for the performance evaluation of the routing protocols being used by the sensor nodes for habitat monitoring. If the temperature of a node becomes too high it might have a considerable impact on the behavior of the species being monitored. It can easily be observed from Figure 6.3 that the LTR produces a much smaller maximum rise in temperature than that of SHR.

SHR always tries to route packets using the shortest path and thus does not consider the temperature rise of the router nodes while routing the packets from source to
destination. Thus sometimes the packets may be forwarded to the destination using the same intermediate nodes repeatedly and hence increases their temperature significantly.

A node using LTR on the other hand forwards the packets to the coolest neighbor and thus achieves load balancing in the network. This leads to better performance in terms of maximum temperature rise in any node.

The packet delivery delay is important for some applications of habitat monitoring, e.g. sending streaming video of the habitat to the base station. It can be seen in Figure 6.4 that SHR produces the lowest delay. Although LTR produces slightly higher delay than SHR, the increase in delay is definitely within acceptable limits. The increase in delay while using LTR is negligible as compared to the improvement obtained in terms of reducing the average temperature of the nodes and the maximum temperature rise of any node.
Power consumption is very important for any sensor network applications. As shown in Figure 6.5, LTR has only slightly higher power consumption than the SHR.
protocol. However both protocols consume low power and thus have very efficient power usage.

Both the LTR and SHR protocols do not drop any packets while routing. The SHR protocol does not drop any packet as it routes the packets using the shortest path. The LTR protocol does not result in packet drop as the data packets need to reach any one of the peripheral nodes using the protocol.

![AVERAGE TEMPERATURE RISE VS. TIME](image)

**Figure 6.6: Average temperature rise vs. Time**

It can be seen from Figures 6.6, 6.7 and 6.8 that for a given packet injection rate in the network or for a fixed sensing rate of the nodes, the network attains a steady state in terms of the average temperature rise of the nodes, the maximum temperature rise of any node and the average packet delivery delay. This validates the fact that the simulation results are stable.

Figure 6.9 shows that the average power consumption of the nodes increases with time by using both the LTR and SHR algorithms for routing.
Figure 6.7: Maximum temperature rise vs. Time

Figure 6.8: Average packet delivery delay vs. Time
6.5. Conclusion

In this chapter a thermal aware routing protocol namely LTR has been presented, which can be used for habitat monitoring applications. The basic idea behind the protocol is similar to the one proposed in Chapter 2 for Biomedical Sensor Networks. The sensor network running the LTR protocol will produce much less heat than those using other routing protocols like the Shortest Hop Routing protocol, that are not thermal aware and do not try to minimize the temperature rise of the nodes. Thus the nodes running the LTR protocol will operate inconspicuously and will not destroy the ecological balance of the habitat. In our future work, we plan to address issues regarding the fault tolerance of the networks deployed to monitor animal habitats and the use of protocols to increase the operating lifetime of the nodes deployed.
CHAPTER 7
TOPOLOGY ADAPTIVE GOSSIP ALGORITHM FOR WIRELESS SENSOR NETWORKS

7.1. Introduction

Many sensor network applications require a reliable network-wide broadcast service to disseminate data throughout the network. The simplest broadcast algorithm that guarantees that nearly 100% of the nodes will receive the broadcast packets is controlled Flooding. In controlled Flooding, nodes forward previously unseen packets to all the neighbor nodes. Flooding is not suitable for being used in sensor networks as it causes a lot of redundant transmissions in the network, leading to unnecessary energy consumption and network congestion [34-35]. In a network that uses Flooding, the packets also incur the least average packet delivery delay as it explores all the possible paths.

In probabilistic broadcast approaches, called Gossip [36-37] the nodes forward packets with certain predefined probability, \( pgossip \). The basic idea behind Gossip is that if \( pgossip \) is properly chosen, the entire network receives the broadcast data with a fairly high probability. However in certain topologies even a fairly high value of \( pgossip \) does not always ensure that all the nodes will receive the broadcast packets.

For example, let us consider the topology in Figure 7.1. In this topology to ensure that all the nodes receive the broadcast packets, the node A should gossip with node B with a probability \( pgossip=1.0 \). However if any other value of \( pgossip \) is used (even if it is as high as 0.90), there will be instances when the nodes B, C and D will not receive the broadcast data packets. A value of \( pgossip=1.0 \) makes Gossip perform similar to Flooding. So that is not at all desirable.
Most routing protocols developed for wireless ad-hoc and sensor networks do not consider the existence of unidirectional or asymmetric links i.e., a signal transmitted by A is received at B, but not vice versa. Such links are often found in the real deployment of wireless ad-hoc networks as shown in recent experimental studies [38-39].

There are some broadcast routing algorithms that assume that all the links in the network are bidirectional. So they are not suitable for being used in a realistic application of sensor network having unidirectional links.

In this chapter we propose a broadcast routing protocol that ensures that nearly 100% of the nodes in the network will receive the broadcast data packets. The network might contain unidirectional links. The proposed TAG algorithm also ensures low average and maximum packet delivery delay. TAG also reduces the buffer requirement at each node and prevents network congestion.

7.2. Proposed Algorithm

The proposed Topology Adaptive Gossip (TAG) algorithm consists of two phases, namely the “Topology discovery” phase and the “Data routing” phase.
• **Topology discovery:**
  - A “topology check” packet is flooded in the network by the source node.
  - If a node receives a “topology check” packet, it forwards the packet to the neighbors, except the neighbor from which it received the packet.
  - Each node has a counter $DPC_i$ (Duplicate Packet Counter) for each of its $i$-th neighbors. This keeps track of the number of times the “topology check” packet is received from each of the $i$-th neighbors.
  - Each time a node receives a “topology check” packet from a neighbor $i$, the corresponding $DPC_i$ is incremented by 1.

• **Data Routing:**
  - If a node has packets to route and $DPC_i == 0$, the node gossips with neighbor $i$ with a probability $pgossip = 1.0$. The value of $DPC_i$ can be 0 due to two different reasons:
    - There is no other path from the source node to the $i$-th neighbor except through this node.
    - This node can reach the $i$-th neighbor but the $i$-th neighbor cannot reach this node as in wireless sensor networks unidirectional links might be present.
  - To ensure that the $i$-th neighbor receives the data packets if $DPC_i == 0$, the node assumes that there is no other path for the packets to reach neighbor $i$ from the source node and hence it always forwards the packets to the neighbor $i$. 
If the $DPC_i$ for the $i$-th neighbor is $> 0$, the node gossips with neighbor $i$ with a probability $pgossip = (1/ (DPC_i)) + C$.

The “Topology discovery” phase is run at certain appropriate intervals to detect changes in the topology due to failure of nodes.

### 7.3. Simulation

The simulation program was developed in C. We performed the simulation over different network topologies and compared the performance of the proposed TAG algorithm with the flooding based broadcast algorithm and the probabilistic gossip algorithm.

The metrics used for comparison of the algorithms are:

A. **Average reception percentage**: The average reception percentage gives a measure of the reliability of the broadcast routing algorithm. It is the average percentage of nodes receiving a broadcast packet.

B. **Average number of packet forwards**: The average number of packet forwards gives a measure of the overhead of the broadcast routing algorithm. It is the average number of nodes to which any node of the network forwards a broadcast packet.

C. **Average packet delivery delay**: The average packet delivery delay gives a measure of the average delay encountered to deliver a broadcast packet to the nodes in the network.

D. **Maximum packet delivery delay**: The maximum packet delivery delay gives a measure of the maximum delay encountered to deliver a broadcast packet from the source to the nodes in the network.
E. Minimum buffer size required at each node: The minimum buffer size required at each node to prevent packet drops gives a measure of the network congestion and is an important metric for comparison the performance of different broadcast routing protocols.

The simulation was run over four different network topologies:

A. 50N_S : A sparsely connected network of 50 nodes (Figure 7.2).
B. 50N_D : A densely connected network of 50 nodes (Figure 7.3).
C. 100N_S : A sparsely connected network of 100 nodes (Figure 7.4).
D. 100N_D : A densely connected network of 100 nodes (Figure 7.5).

![Figure 7.2: A sparsely connected network of 50 nodes](image1)

![Figure 7.3: A densely connected network of 50 nodes](image2)
Figure 7.4: A sparsely connected network of 100 nodes

Figure 7.5: A densely connected network of 50 nodes

Figure 7.6 compares the average packet reception percentage when using the different routing algorithms in the four different topologies. For all the topologies Flooding achieves an average packet reception percent of 100. The proposed TAG algorithm also achieves nearly 100 percent “average reception percentage”. When using Probabilistic Gossip with a probability $pgossip=0.20$, the average reception percentage is pretty low. However if it is used with a probability $pgossip=0.80$, it achieves better “average reception percentage” although still it is significantly lower than that achieved by Flooding or TAG algorithms.
Figure 7.6: Average packet reception percentage for different network topologies

Figure 7.7: Average number of packet forwards per node for different network topologies

Figure 7.7 compares the average number of packet forwards per node per packet. It can be seen that Flooding has a very high average number of packet forwards. TAG
significantly lowers average number of packet forwards, especially in the densely connected topologies, namely 50N_D and 100N_D. Probabilistic gossip has low average number of packet forwards per node.

As seen from Figures 7.6 and 7.7, the TAG algorithm achieves very high reliability as it has an average reception percentage of nearly 100. On the other hand unlike flooding it has a low overhead as it reduces the unnecessary packet forwards and keeps the average number of packet forwards low. Although Probabilistic gossip keeps the average number of packet forwards low, it is not suitable for being used in applications requiring a high reliability as even with probability \( pgossip=0.80 \), the average reception percentage is not good. If the value of \( pgossip \) be increased beyond 0.80 the Probabilistic gossip loses its significance and performs similar to flooding.

![Average Packet Delivery Delay](image)

**Figure 7.8: Averagge packet delivery delay for different network topologies**
The average and maximum packet delivery delays of the TAG algorithm and Flooding have been compared to ensure that the packet delivery delay is not significantly increased when using TAG algorithm.

Figures 7.8 and 7.9 show that for all the topologies the average and maximum packet delivery delays when using TAG is only marginally higher than those when using Flooding.
Figure 7.10 compares the minimum buffer size required at each node to prevent packet drops when using the different broadcast routing algorithms. It can be seen that in all the four topologies, use of Flooding requires the nodes in the network to have the largest buffer size followed by the Probabilistic Gossip (with $pgossip=0.80$). The proposed TAG algorithm has the least buffer space requirements and is thus less prone to congestion.

### 7.4. Conclusions

The proposed Topology Adaptive Gossip (TAG) algorithm is a very reliable broadcast routing algorithm for use in networks having unidirectional links and involves a very low overhead. TAG also has very low average and maximum packet delivery delays and a low buffer size requirement.
CHAPTER 8
A TUNABLE CROSS-LAYER CONGESTION REDUCING MEDIUM ACCESS CONTROL (CRMAC) PROTOCOL FOR WIRELESS NETWORKS

8.1. Introduction

Over the past few years network congestion has been identified as a major problem in wired networks as well as ad-hoc wireless and sensor networks [46], [51-53]. In wireless adhoc networks, congestion often sets in due to the bursty nature of data traffic resulting in large number of packet drops due to buffer overflow in the network layer and an increase in packet delivery latency [46], [54]. Wireless Sensor Networks (WSNs) often run data-gathering applications in which data gathered by a large number of sensors need to be delivered to a sink node through a multi-hop wireless network. Such networks often experience severe network congestion due to the limited buffer capacity of the sensor nodes and limited capacity of the shared wireless medium [53]. Congestion leads to a drop in the overall network throughput and energy waste in the highly energy constrained wireless sensor nodes [54-55]. If the sensor network is used for certain event detection applications, the loss of packets due to buffer overflow might also hamper the event detection reliability [53].

Most current research on congestion control focus mainly on the development and analysis of end-to-end control schemes by modifying existing schemes used in wired networks and adapting them to wireless domain [51]. Some other research focus on the development of new routing and Medium Access Control (MAC) protocols that aim at reducing congestion in the network [55-57].
Hop-by-hop congestion control schemes using feedback mechanism have also been widely studied. Feedback about the congestion state of a node based on its buffer queue length is sent to the preceding node, which reacts to the congestion information by adjusting its transmission rate [58]. If the paths experience congestion persistently, the hop-by-hop backpressure eventually reaches the source and allows them to throttle the transmission rate. The authors in [51] have mentioned that such hop-by-hop congestion control schemes react much faster than end-to-end schemes. However, such schemes might not be very suitable for multi-hop wireless networks since the backpressure needs to propagate through multiple hops through an already congested network before the source can throttle its transmission rate.

Contention based Medium Access Control (MAC) methods constitutes one of the major sources of network congestion [53], [58]. Use of contention based MAC protocols like 802.11 in multi-hop wireless networks makes the network congestion even worse.

![Figure 8.1: Simple topology consisting of 5 nodes in the same collision domain](image)

A standard Carrier Sense Multiple Access (CSMA) MAC, like 802.11 gives all the nodes in the same collision domain an equal chance to transmit. This approach might
lead to congestion in high fan-in scenarios. To have an insight of the problem let us consider the simple topology in Figure 8.1. The nodes B, C and D transmit data packets to sink S through intermediate node A. All the nodes are in the same collision domain. Let us assume that B, C and D operate in saturation condition, i.e. they always have available data to send to A. A has a limited buffer space. When a standard CSMA MAC is used in the scenario, nodes A, B, C and D will get an equal opportunity to transmit data frames. So on average when B, C and D will get a chance to transmit 1 packet each, node A will also get a chance to transmit one data packet to S. If the network operates for a long time, the packet buffer of node A will become full, leading to congestion and packet drops, since after each round of transmission there will be an extra backlog of (3-1)=2 packets at node A’s packet buffer.

To solve the congestion problem in multi-hop wireless networks, we propose a cross-layer MAC protocol, Congestion Reducing Medium Access Control (CRMAC). The main rationale behind this approach is that in a multi-hop wireless network, the MAC protocol has a large impact on the overall network throughput [53], [59]. In most cases the MAC layer throughput limits the Network layer throughput.

The need for cross-layer protocol designs has been emphasized by the authors in [52], [53]. It might not be sufficient to optimize the performance of the network or the MAC layers in isolation.

To solve the congestion problem, we resort to the idea of prioritized access [46], [60]. The basic idea of CRMAC is that in the MAC layer frame transmission, the congested nodes will be given priority over the others in the same collision domain. To achieve this, the MAC layer will require cross-layer information about the status of the
buffer from the Network layer. However, the buffer status alone cannot be an accurate measure of the network congestion as it is purely local information and cannot reflect the global congestion status. So a metric that gives the measure of the congestion status of other nodes in the same collision domain is necessary. Collision history of a node can be such a metric. The more the number of collisions, the more congested the network is and vice versa [53]. The proposed CRMAC protocol takes into account both the buffer status of the node and the collision history when giving the congested nodes a prioritized medium access.

Since ad-hoc wireless networks and wireless sensor networks are being used for a wide range of applications, the protocols used in such networks might have different application-specific performance requirements. Some wireless network applications might tolerate some energy wastage due to collisions in the MAC layer but require the network to have a very high throughput and low latency. Some other applications might be delay tolerant but are adversely affected by energy wastage due to collisions in the MAC layer. So the protocol designed should be tunable to different application-specific performance requirements without the need to make significant changes in the protocol design. The proposed CRMAC protocol can be tuned to various performance requirements, required by different wireless network applications.

8.2. Proposed protocol

The CRMAC protocol is similar to the 802.11 protocol with respect to using RTS and CTS messages to prevent the hidden node problem [58], but differs in the way the backoffs are generated. Instead of using a binary exponential backoff mechanism, as used in 802.11 [58-59], the CRMAC protocol uses information about the remaining buffer
space from the routing layer, to give transmission preference to the nodes that are experiencing buffer congestion over those that have less congested buffers.

The CRMAC needs to maintain some additional information to help it in the generation of the appropriate backoff value.

A. **Buffer capacity (B)** – This gives a measure of the number of packets that the node can buffer.

B. **Remaining buffer capacity (δ)** – This information is provided to the MAC layer by the Network layer and is a measure of the buffer congestion at that node. The less the remaining buffer capacity, the more severe is the node congestion.

C. **Collisions (α)** – This variable keeps track of the number of collisions encountered when the node contends for the shared medium to send a particular frame. The value of α is reset to 0 after each successful frame transmission.

D. **Collision history (β)** – This gives a measure of the contention for the shared medium and hence the congestion status of the collision domain. It is updated after every successful transmission of a frame, before α is reset. Collision history is calculated as $\beta = (\beta \times \mu) + (\alpha \times (1-\mu))$ with $\mu$ set to 0.80. Thus more weight is given to the past history and less weight to the information obtained from the recent past. This is done to prevent the value of collision history from rapidly oscillating.
E. Randomizer window \( (W) \) – This is defined as \( W = \min \left( 2^n, 4 \right) \). Thus the randomizer window grows from 1 to 4 with increasing number of collisions and remains at 4, until the frame is successfully transmitted.

The basic idea behind the CRMAC protocol is that the frame transmission backoff generated should be directly proportional to the remaining buffer size, i.e. the lesser the remaining buffer size, the lesser should the backoff generated be. Also the node should increase the backoff if the node experiences large number of RTS collisions while contending to send a frame or when the node had experienced large number of collisions in the past.

In CRMAC protocol, whenever any node has a packet to transmit, it generates the backoff as follows:

\[
\text{Backoff} = \text{rand}(0,1) \times W + \left\{ \left( \frac{\delta}{B} \right) \times (\alpha + \beta) \times C \right\} \tag{1}
\]

\[
\beta = \left[ \beta \times \mu \right] + \left[ \alpha \times (1 - \mu) \right] \tag{2}
\]

In the above expression, \( \text{rand}(0,1) \) generates a random number between 0 and 1. \( \text{rand}(0,1) \) multiplied with \( W \) generates a random number between 0 and the value of \( W \). The rationale behind adding this when generating the backoff is that there might be two nodes in the same collision domain that have similar buffer conditions and collision history. In that case the random number will help reduce the collisions encountered when two such nodes try to transmit frames.

In equation (1), the generated backoff increases with increase in \( (\delta/B) \), the remaining buffer capacity, expressed as a fraction of the total buffer capacity. The
fraction has been considered instead of $\delta$ alone to prevent the generated backoff from becoming very large, when using large buffers at the nodes.

It can be also observed that the backoff generated by a node is directly proportional to the sum of $\alpha$ and $\beta$, which gives a measure of the global congestion status of the collision domains, that the node belongs to. $\beta$ is updated after every successful transmission of a frame, before $\alpha$ is reset.

In the backoff generation, the constant $C$ plays a very significant role. By setting $C$ to an appropriate value the performance of the CRMAC protocol can be tuned to adapt to the requirements of various applications. Let us consider the following two applications:

A. *Emergency wireless networks*: The wireless networks deployed needs to operate in certain emergency situation for a short duration and needs to provide a high throughput and low packet delivery latency. The networks can have RTS collisions and retransmissions due to packet drops as long as throughput of the network is not compromised. For wireless networks running this type of application, the constant $C$ is set to a small value (<1).

B. *Wireless networks for monitoring purposes*: The wireless networks deployed for monitoring applications usually need to operate for a long time and so cannot tolerate energy waste due to packet retransmissions caused by dropped packets and large number of collisions in the network. On the other hand such applications also need fairly good network throughput although they can tolerate some packet delivery latency. For the wireless networks running this type of application, the constant $C$ is set to a large value (>1).
8.3. Simulation Results

The simulation program has been developed in C++. We performed the simulation over an adhoc wireless network topology consisting of 100 nodes, deployed for the data gathering emergency or monitoring application and compared the performance of the proposed protocol (CRMAC) with that of the 802.11 and FairMAC. As in [1], the network is assumed to have multiple overlapping collision domains. A node can communicate with its neighbors through the shared wireless medium. In the topology shown in Figure 8.2, such pairs of communicating nodes are connected by links. The nodes in the network are homogeneous and contain buffer of same size. The network uses a RTS-CTS-DATA-ACK based scheme like 802.11 to prevent the hidden-node problems. All the nodes are assumed to inject data-packets in the network at a certain fixed rate. The goal of the network is to route the data packets to the sink, located at the root of the data gathering tree, through the multi-hop wireless network. The FairMAC protocol is similar to the 802.11 protocol but has a fixed sized contention window, equal to the maximum contention window size of 802.11. We assume that, for all three protocols, the transmission of one frame between the MAC layers of two nodes results in transmission of one packet between their network layers.

The metrics used for the comparison of the protocols are:

A. *Percentage packets dropped by the network*: This is measured by \[\frac{\text{Number of packets dropped due to buffer overflows}}{\text{Total number of packets injected in the network}}\]. The percentage of packets dropped in the network is a measure of the network congestion. A high value of the percentage packets dropped might result in decrease in the overall network throughput.
B. **Network throughput**: The throughput of the network is defined as the number of data packets delivered to the sink per unit time (slot-time has been taken as the unit of time for this simulation). The throughput of the network is a direct measure of the amount of congestion in the network. As the network becomes more and more congested, throughput drops. Throughput of a network is also one of the most important performance metrics as the overall performance of any application running on the network depends on it.

C. **Average packet delivery delay**: Average packet delivery delay is measured by the average amount of time needed to deliver a packet from the source to destination and also gives a measure of the network congestion. Ad hoc networks deployed for emergency situations are sensitive to packet delivery delay.

D. **Maximum packet delivery delay**: Maximum packet delivery delay is measured by the maximum amount of time needed to deliver a packet from the source to destination.

E. **Number of RTS collisions**: Collisions in most cases tend to decrease the throughput and increase the packet delivery latency. On the other hand, very few or no collisions might mean that the network is underutilized. In case of sensor networks, that need to operate for a long time with limited power supply, collisions are not desired as they result in unnecessary energy waste.

F. **Average number of RTS collisions per data frame**: This gives a measure of the average time spent by a frame in collisions. As long as the average number of
collisions per frame remains low, a large number of RTS collisions will only marginally affect the packet delivery delay.

We have demonstrated the performance of the proposed CRMAC protocol and its tunable nature by considering two different application scenarios having different performance requirements. The packet injection interval is assumed to be 75, i.e. after every 75 simulation intervals each node injects one data packet in the network, to be routed to the sink. All simulation results are given with 95% confidence intervals using the batch-means method with 50 batches (confidence intervals are shown as vertical bars on the plotted graphs).

Figure 8.2: 100 nodes connected in tree topology, with sink at the root

8.3.1. Scenario A

The deployed network is being used in an emergency situation. To achieve the desired performance, the value of C is set to 0.4. The results obtained are shown below.
Figure 8.3: Percentage of injected packets dropped by the network, when using three different MAC protocols and different buffer capacities at the nodes.

Figure 8.4: Network throughput when using three different MAC protocols and different buffer capacities at the nodes.

Figure 8.3 clearly shows that CRMAC performs better than 802.11 in terms of number of packets dropped. Also the positive effect of buffer increase is more when
CRMAC is used, than when using the other two protocols. However for very small buffer size, FairMAC performs better then CRMAC.

From Figure 8.4, it can be seen that when CRMAC is used, the network throughput achieved is more than double that when 802.11 is used and more than six times that when FairMAC is used. Also with increase in buffer size, CRMAC unlike the other two protocols shows a steady increase in throughput. The reason for FairMAC having a very low throughput is that it has a relatively large, fixed sized contention window, which causes the nodes to generate large backoffs, independent of the network contention status.

![Average Packet Delivery Delay for Different Buffer Size](image)

**Figure 8.5:** Average packet delivery delay when using three different MAC protocols and different buffer capacities at the nodes
Figure 8.6: Maximum packet delivery delay when using three different MAC protocols and different buffer capacities at the nodes

Figures 8.5 and 8.6 show that both the average and the maximum packet delivery delays or latencies are minimized when CRMAC is used. Another interesting observation that can be made is that with increase in buffer size, for all the protocols, the average and maximum packet delivery delays increase. This might be attributed to the fact that if the nodes have large buffers, the packets will stay in the buffers of the intermediate nodes in the path for a long time, resulting in large packet delivery delays.
Figure 8.7: Number of RTS packet collisions when using three different MAC protocols and different buffer capacities at the nodes.

Figure 8.8: Average number of RTS packet collisions per data frame when using three different MAC protocols and different buffer capacities at the nodes
Figure 8.7 shows that the number of RTS frame collisions is highest when using CRMAC. Use of FairMAC results in least number of RTS collisions. The number of RTS collisions when using 802.11 is lower than that when CRMAC is used, but higher than the same when FairMAC is used. The good thing about CRMAC is that when using CRMAC, the number of RTS collisions decreases progressively with the increase in the buffer capacity of the nodes. However this is not a problem as the application running on the network in scenario A can tolerate some RTS collisions as long as it does not affect the throughput. The gain achieved by the cross layer design in decreasing packet drops more than offsets the negative effect of increased collisions. From Figure 8.8 it can be seen that although the total number of RTS collisions when using CRMAC is high, the average number of RTS collisions per frame is low, only slightly higher than that when 802.11 is used. Thus in CRMAC, the impact of RTS collisions on packet delays is minimal for two reasons: i) the size of the RTS frame is very small, and ii) the backoff value computed by equation (1) does not grow at the same fast growth rate of the backoff used in 802.11.

The above simulation results show that CRMAC can be tuned to perform well when used in “emergency wireless networks”. The network using CRMAC algorithm delivers packets to the sink with very low latency and achieves a very high network throughput and performs much better than FairMAC or 802.11. A low value of \( C \) results in small values of backoff on the average if the buffers in the network layer become congested. This helps the network achieve a high throughput. Due the same reason the number of RTS collisions is also high.
8.3.2. Scenario B

The deployed network is being used in a monitoring application. To achieve the desired performance, the value of $C$ is set to 3.0. The results obtained are shown below:

![Percentage of packets dropped for different buffer sizes](image)

Figure 8.9: Percentage of injected packets dropped by the network, when using three different MAC protocols and different buffer capacities at the nodes

Figure 8.9 shows that CRMAC performs better than 802.11 in terms of number of packets dropped. The number of packets dropped by the network decreases drastically with the increase in buffer capacity, when CRMAC is used. When a fairly large buffer is used, use of CRMAC results in very few packet drops.

From Figure 8.10 it can be observed that the network throughput achieved when using 802.11 or CRMAC is much higher than that when FairMAC is used. CRMAC achieves a little higher throughput than 802.11.
Figure 8.10: Network throughput when using three different MAC protocols and different buffer capacities at the nodes.

Figure 8.11: Average packet delivery delay when using three different MAC protocols and different buffer capacities at the nodes.
Figures 8.11 and 8.12 show the average and the maximum packet delivery delay for the three protocols. The average delays for CRMAC lower than that of FairMAC but is higher than that of 802.11. However this should not be a problem in scenario B as the monitoring applications are not usually delay sensitive.
Figure 8.13: Number of RTS packet collisions when using three different MAC protocols and different buffer capacities at the nodes.

Figure 8.14: Average number of RTS packet collisions per data frame when using three different MAC protocols and different buffer capacities at the nodes.

Figure 8.13 shows that the number of RTS collisions encountered in MAC layer when using CRMAC protocol is much lower than that when using 802.11. The number of
collisions also decreases significantly with increasing buffer capacity. A similar trend can also be seen in Figure 8.14, which shows the average number of RTS collisions per frame.

The above simulation results show that CRMAC can also be tuned to perform well in wireless sensor networks operating for a long time. The network using CRMAC algorithm delivers packets to the sink with good throughput, low packet drops and acceptable number of RTS collisions and performs much better than FairMAC or 802.11. A large value of $C$ results in considerably large values of backoff on the average and helps reduce number of collisions.

We also studied the behavior of the MAC protocols under different network loads and compared their performance. The network load has been varied by varying the packet injection interval for the nodes from 200 to 25 simulation intervals, in steps of 25. All the nodes are assumed to have a buffer capacity of 80. In the results CRMAC-1 denotes the CRMAC protocol with $C=0.4$ and CRMAC-2 denotes the CRMAC protocol with $C=3.0$.

It can be seen from Figure 8.15 that none of the four MAC protocols produce any packet drops when the network load is low. At high network loads use of all the four protocols result in packet drops. However FairMAC and 802.11 produce much higher packet drops than CRMAC-1 and CRMAC-2. Use of CRMAC-2 results in less number of packet drops than when CRMAC-1 is used as is desired.
Figure 8.15: Percentage of injected packets dropped by the network with increasing network load when using three different MAC protocols

Figure 8.16: Network throughput with increasing network load when using three different MAC protocols
Figure 8.16 shows that use of FairMAC results in lowest throughput, whereas the throughput obtained when using CRMAC-1 is the highest. When using FairMAC or 802.11, the throughput initially remains constant with increasing network load and then starts to drop when the load is increased further. In case of CRMAC-1 and CRMAC-2 the throughput increases with increase in network load, reaches a peak throughput and then starts to drop.

![Average Packet Delivery Delay with Increasing Network Load](image)

**Figure 8.17: Average packet delivery delay with increasing network load when using three different MAC protocols**

Figure 8.17 shows that the average packet delivery delay is highest when FairMAC is used and least when CRMAC-1 is used. With increase in network load beyond a certain limit, all the protocols show an increase in average packet delivery delay, but the delay increases very sharply when FairMAC or 802.11 is used.
Figure 8.18: Maximum packet delivery delay with increasing network load when using three different MAC protocols

Figure 8.19: Total number of RTS packet collisions with increasing network load when using three different MAC protocols

Figure 8.18 shows that the maximum packet delivery delay is least when using CRMAC-1. Although the network using CRMAC-2 experiences a sharp rise in maximum
packet delivery delay at a moderate network load, the delay stabilizes when the load increases further and eventually performs better than FairMAC and 802.11 at high load conditions.

Figure 8.19 shows that under all load conditions use of CRMAC-1 results in highest number of RTS collisions, followed by 802.11, CRMAC-2 and FairMAC. The number of RTS collisions per data frame also follows a similar trend, as evident from Figure 8.20.

![AVGERAGE NUMBER OF RTS COLLISIONS PER DATA FRAME WITH INCREASING NETWORK LOAD](image)

**Figure 8.20:** Number of RTS packet collisions per data frame with increasing network load when using three different MAC protocols

Thus it can be seen that the proposed CRMAC protocol can be tuned to perform as desired under all network load conditions.

### 8.4. Conclusion

This chapter presents a cross-layer medium access control protocol, CRMAC that aims at reducing buffer congestion and packet drops in the network layer. CRMAC uses
buffer status information from the Network layer and MAC layer collision history to give prioritized channel access to congested nodes. The CRMAC protocol is easily tunable to different application-specific performance requirements, which make it usable for different wireless adhoc and sensor network applications.
CHAPTER 9
HOTSPOT PREVENTING ADAPTIVE ROUTING ALGORITHM FOR NETWORKS-ON-CHIP

9.1. Introduction

The Network on Chip (NoC) is made up of routers that are connected to each other by a point-to-point network and provides a backbone for the different components of the chip to communicate [18-22]. The network allows the components on the chip to communicate concurrently using the same bus. This gives an advantage over using bus-based architecture in which a bus allows only one communication at a time [23].

The power density in microprocessors has doubled every three years in the recent past [24-25]. Since a large part of the consumed energy is converted into heat there has been an exponential rise in neat density [25]. Heat generated in the microprocessors needs to be removed from the microprocessor surface by the cooling system. So the cooling costs are also rising exponentially [25].

Thermal conditions of chips are greatly impact chip reliability and performance. The temperature of a chip is affected by both processing and communication components. Since the on-chip network backbone consumes significant portion of the chip power budget, it has a large influence over the thermal behavior of the chip.

Formation of hotspots can lead to performance deterioration and often failure of the chips. Large temperature variations and hotspots account for over 50% of electronic failures [26]. So formation of hotspots needs to be prevented in order to prevent chip failures [27].
In a packet switched network the routing protocol decides the paths that packets should take to go from a source to destination. The routes chosen for delivering packets can greatly affect the formation of hotspots in the network.

Currently most of the routing algorithms for NoC depend on static routing mechanisms [18]. Such routing schemes are mainly based on shortest path information stored in the routing tables in the routers and hence do not adapt to changes of topology due to the failures of routers and links.

[19] proposes a dynamic routing algorithm that adapts to the changes in topology and router failures. The proposed dynamic routing mechanism for NoCs reconfigures the network and rebuilds the routing tables when a link or router failure is detected. The algorithm uses Dijkstra’s Shortest Path First algorithm to build the routing tables. However the algorithm does not take into consideration the thermal behavior of the on-chip network.

[27] proposes a scheme for placing components on a chip to achieve even temperature distribution over the entire chip. But the scheme is based on a traditional bus-based architecture and is not suitable for use in NoCs.

The problem of high temperature zones in NoCs have been addressed by [25] at the micro-architecture level. The paper emphasizes on the importance of thermal modeling at the architecture level as a step towards to research in temperature-aware computing.

A thermal modeling of NoCs has been done in [24]. The paper points out to the fact that networks contribute to a large part of the heat produced in chips where it is deployed and hence thermal aware protocols are required for networks on chip.
The first work that addresses the problem of formation of hotspots is [28]. In this paper the workload mapping is changed dynamically at runtime to prevent the formation of hotspots. The workload mapping is done in such a way that hotspot inducing computations are shifted or migrated across the chip to make the thermal profile of the chip uniform. But such dynamic reconfigurations have very large overhead and deteriorate performance.

[29] and [30] address the necessity of temperature aware routing schemes in 3D ICs. However the proposed solutions work on a bus-based architecture and are not suitable for being used in a packet switched NoC.

The most commonly used routing algorithm is the Shortest Hop Routing (SHR) algorithm where the packets take the shortest paths (minimum number of hops) to reach the destination. However the shortest path algorithm does not take into account the thermal condition of the network or the occurrences of hotspots when routing packets.

To our knowledge the proposed Hotspot Preventing Adaptive Routing (HPAR) algorithm [5] is the first algorithm that takes into account the thermal condition of the network on-chip when routing packets and thus prevents the formation of hotspots.

9.2. Proposed Algorithm

The Hotspot Preventing Adaptive Routing (HPAR) algorithm consists of two main phases, the setup phase and the routing phase.

- **Setup Phase:**
  - The components or modules attached to the routers on the chip have unique *module_ids*. The routers exchange information about the *module_ids* and the associated routers.
The routers exchange information and build the routing tables, which contain the shortest path information. Each router on the chip has a unique router_id. The routing tables are refreshed at regular intervals.

- The routers exchange information about their initial temperature.

- **Routing Phase:**
  - Each router handles two types of packets, those that originate at the modules connected to the router and those that are coming from other routers to be routed to destination. Both types of packets are treated in the same way by the router.
  - Each router tries to route the packets in such a way that the packets get delivered to the destination using the minimum number of hops, i.e. the shortest path, unless there is a hotspot in the path.
  - If the module_id of the destination of the packet is associated with the router itself, the packet is delivered to the component or module.
  - Else if the module_id of the destination of the packet is associated with a neighboring router, the packet is forwarded directly to the corresponding router.
  - Else if the next-hop in the shortest path to the destination has a temperature less than or equal to the temperature of the router plus a threshold, the packet is forwarded to the next-hop router in the shortest path to the destination.
  - Else if the next hop in the shortest path to the destination has a temperature greater than the temperature of the current router plus a threshold:
- The router predicts that the path under consideration goes through a hotspot.
- The router tries to bypass the hotspot and route the packet.
- The router forwards the packet to the neighbor router that has the least temperature or the “coolest neighbor”.
- Each packet maintains a small list of routers it has most recently visited. If the “coolest neighbor” is already there in the list of recently visited routers for the packet, the packet is forwarded to the neighbor having second lowest temperature among the neighboring routers and so on.
  - Experiments show that with the increase of the packet injection rate in the system, the average temperature of the nodes increases. So does the variation in temperature among the neighboring nodes. The value of the threshold is dynamically set based on the local load (packets routed by the router over a past time window) and the temperature of the neighboring routers. Both factors are given equal weight in calculating the threshold value.
  - The temperature of a router gives a measure of the number of packets routed by it over past window and hence the load handled by the router. Hence the threshold value depends on two components:
    - Component C1: A function of the average temperature of the neighboring routers.
    - Component C2: A function of the router’s own temperature.
  - C1 and C2 are found out by experiments.
- \( C1 = k1 \times \sqrt{\text{avg}_n} \)
- \( C2 = k2 \times \sqrt{\text{temp}_n} \)

- \( k1 \) and \( k2 \) are constants, \( \text{avg}_n \) is the average temperature of the router’s neighbors and \( \text{temp}_n \) is the temperature of the router.
- The threshold is calculated giving equal weights to the two components. \( \text{threshold} = 0.5 \times C1 + 0.5 \times C2 \).
- The routers periodically exchange their temperature information with the neighbors.

9.3. Simulation

The simulation program has been developed in C. We performed the simulation over different on-chip network topologies and compared the performance of the proposed algorithm (HPAR) with that of the Shortest Hop Routing algorithm (SHR).

The metrics used for the comparison of the algorithms are:

A. Maximum temperature rise of any router node of the NoC: The maximum temperature rise of any node is an important metric that needs to be compared as it is directly associated with the formation of hotspots, failure of the chip and the cooling costs for a chip.

B. Average packet delivery delay in the NoC: Average packet delivery delay is a measure of the average delay experienced by a packet before it reaches its destination. This is an important metric as the QoS performance of the chip depends on this.

The simulation was run over three different on-chip network topologies. The two dimensional mesh and the mesh-torus are the most commonly used Network-on-Chip
topologies. Three-dimensional integrated circuits (3D ICs) provide solutions to deal with problems of scalability and increasing interconnect costs for next generation chips [31-33].

A packet arrival rate or packet injection rate of 0.5-1.0 packets per simulation interval has been defined as low packet arrival rate and the system is said to be working under a light load, whereas a packet arrival rate of 3.5-4.0 packets per simulation interval is defined as high packet arrival rate and the system is said to be working under heavy load.

Each simulation run generated 2000 data packets that are routed through the system. The source and destination of the packets are randomly generated and packet arrival follows a Poisson distribution. Each time a router receives a packet to route, its temperature increases by 0.1 units. The cooling rate of the system (for all the routers) is assumed as 0.1 units per simulation interval. It has been assumed that none of the data packets gets dropped by routers.

9.3.1. Two dimensional mesh topology of 4x8 routers

![Figure 9.1: 2D Mesh of 4x8 routers](image)
Figure 9.2: Maximum temperature rise vs. packet arrival rate in 2D Mesh topology

Figure 9.3: Maximum temperature rise of the router nodes using SHR and HPAR algorithms under light load in 2D Mesh topology
From Figure 9.2 it can be seen the SHR algorithm shows significantly higher temperature rise compared to the HPAR algorithm with increasing packet arrival rate to the system. Figure 9.3 shows that even with a lightly loaded system, the use of SHR algorithm tends to produce hotspots. Figure 9.4 shows that when the system is heavily loaded SHR produces hotspots having extremely high temperatures. HPAR performs much better by removing the hotspots to a large extent.
Figure 9.5: Average packet delivery delay vs. packet arrival rate in 2D Mesh topology

Figure 9.5 shows that the HPAR produces higher packet delivery delay when the system is not lightly loaded. However the increase in delay is not significant as compared to the improvement in performance obtained as far as removal of hotspots is concerned.

9.3.2. Two dimensional mesh-torus topology of 4x8 routers

Figure 9.6: 2D Mesh-torus topology of 4x8 routers
Figure 9.7: Maximum temperature rise vs. packet arrival rate in 2D Mesh-torus topology

Figure 9.8: Maximum temperature rise of the router nodes using SHR and HPAR algorithms under light load in 2D Mesh-torus topology
Figure 9.9: Maximum temperature rise of the router nodes using SHR and HPAR algorithms under heavy load in 2D Mesh-torus topology

Figure 9.10: Average packet delivery delay vs. packet arrival rate in 2D Mesh-torus topology
Figure 9.7 shows that the performance improvement obtained by using HPAR algorithm is much more in the mesh-torus topology than in the mesh topology. The efficiency of HPAR in reducing hotspots in both heavily and lightly loaded networks can be easily seen from Figures 9.8 and 9.9.

Figure 9.10 shows that the average packet delivery delay increases only by a negligible amount for all packet arrival rates when using HPAR algorithm. The reason for this small increase is that the mesh-torus topology has higher connectivity and has more routing options than the mesh topology. Each node in the mesh torus network has four neighbors which increases the flexibility of HPAR in avoiding hotspots.

9.3.3. Three dimensional topology of 27 routers

Figure 9.11: 3D Mesh of 27 routers
Figure 9.12: Maximum temperature rise vs. packet arrival rate in 3D Mesh topology

It can be seen from Figure 9.12 that in a 3D topology use of HPAR algorithm produces much smaller rise in temperature than using SHR. So the costs of cooling are greatly reduced.

As seen in Figures 9.13 and 9.14, HPAR algorithm performs significantly better than SHR algorithm when the routers are connected in a 3D topology. Using SHR produces routers having extremely high temperatures.
Figure 9.13: Maximum temperature rise of the router nodes using SHR and HPAR algorithms under light load in 3D Mesh topology.

Figure 9.14: Maximum temperature rise of the router nodes using SHR and HPAR algorithms under heavy load in 3D Mesh topology.
From Figure 9.15 it can be seen that use of HPAR in place of SHR increases the delay only slightly.

The simulation results obtained by using the SHR and HPAR algorithms in the three topologies can be explained as follows.

In all of the three topologies SHR produces very high temperature in certain router nodes. This is because the SHR forwards the packets along the shortest path and does not consider the temperature rise of the router nodes while routing the packets from source to destination. Thus sometimes the packets may be forwarded to the destination using the same routers repeatedly and hence increasing their temperature significantly. The HPAR algorithm on the other hand bypasses hotspots while routing packets and thus prevents the formation of zones having very high temperatures.

It can be observed from the simulation results that although the HPAR algorithm performs much better than the SHR algorithm in terms of reducing the maximum
temperature rise of the routers, the increase in average packet delivery delay while using HPAR instead of SHR algorithm is very small. This is due to the fact that HPAR does not route packets through arbitrary long paths. Instead it always tries to route packets through the shortest path from the source to the destination, only bypassing the hotspots that appear in the path.

9.4. Conclusion

This Chapter presents a routing algorithm HPAR that can be used in networks on-chip. The algorithm helps prevent the formation of hotspots and hence if used in on-chip networks will reduce the costs of cooling and the chances of failures of chips due to overheating. The main idea behind the proposed algorithm is similar to the Hotspot Preventing Routing (HPR) algorithm proposed in Chapter 3. Thus it can be concluded that on-chip networks have similar design constraints as in-vivo networks of biomedical sensors. The proposed HPAR algorithm is the first of its kind and paves a way towards research in routing algorithms of thermal-aware computing.
CHAPTER 10
CONCLUSION

The thesis has presented different Routing algorithms and Medium Access Control protocols for Wireless Sensor Networks and On-chip Networks. The proposed algorithms have been validated with the help of extensive simulations. The performance of the proposed methods and existing ones has been compared by using appropriate application-specific evaluation metrics.

We have developed different routing algorithms for Biomedical Wireless Sensor Networks, the first one being Adaptive Least Temperature Routing (ALTR) algorithm, described in Chapter 2. The Hotspot Preventing Routing (HPR) algorithm proposed in Chapter 3 tries to remove the shortcomings of the ALTR algorithm. Routing Algorithm for network of homogeneous and Id-less biomedical sensor Nodes (RAIN) discussed in Chapter 4 aims at solving the data routing problem in networks of in-vivo biomedical sensor nodes, lacking unique hardware identifiers. The proposed novel algorithm is the first such algorithm for networks of in-vivo id-less sensors. Finally in Chapter 5 we proposed Biocomm, a cross-layer Medium Access Control and routing protocol co-design for biomedical sensor networks. Extensive simulations show that the cross-layer network architecture helps optimize the overall network performance and it performs much better than the HPR algorithm discussed in Chapter 3.

In Chapter 6 we presented a routing protocol that can be used for habitat monitoring applications.

Chapter 7 described Topology Adaptive Gossip (TAG) algorithm, which can be used for network-wide broadcast in Wireless Sensor Networks.
To reduce network congestion Biocomm also uses an idea similar to the basic idea of the Congestion Reducing Medium Access Control (CRMAC) protocol, described in Chapter 8, i.e. use Network layer buffer congestion information to give higher frame transmission priority to congested nodes in the MAC layer.

Chapter 9 describes a routing algorithm developed for being used in On-chip Networks. The algorithm uses an idea similar to the HPR algorithm proposed in Chapter 3 for being used in in-vivo biomedical sensor networks. So it can be concluded that algorithms used in in-vivo networks and On-Chip networks have very similar design constraints.

Thus in this thesis we have addressed different applications of Wireless Sensor networks and On-Chip Networks and presented efficient MAC protocols and routing algorithms for such networks. Wireless Sensor Networks is no longer a dream and within the next decade it will bring a radical change in our life.
REFERENCES


