COMMUNICATION PROTOCOLS FOR ENERGY CONSTRAINED NETWORKS

TAN HWEE XIAN

NATIONAL UNIVERSITY OF SINGAPORE

2011

COMMUNICATION PROTOCOLS FOR ENERGY CONSTRAINED NETWORKS

TAN HWEE XIAN
(B. Computing (Hons), NUS)

A DOCTORAL THESIS SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
SCHOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE

2011

Acknowledgements

My advisor Professor Mun-Choon Chan has been an invaluable source of guidance and support throughout the research in this dissertation. He has dedicated immeasurable time and effort in honing my research skills, and pushed me to think more critically by constantly challenging my ideas. I am most grateful for his patience and commitment, as well as friendship.

I also wish to extend my sincere gratitude to the following people from National University of Singapore (NUS), who have given me much advice and encouragement during this journey: Professor A. L. Ananda, Professor Wei Tsang Ooi, Dr. Colin Tan, Dr. Ben Leong, Mr. Aaron Tan and Jun Ping Ng.

I am thankful to have wonderful friends in the Communication and Internet Research Lab (CIRL) in School of Computing, NUS - whose encouragement, friendship, laughter and insightful discussions have accompanied me through many long days and nights: Eugene Chai, Binbin Chen, Mingze Zhang, Xiuchao Wu, Tao Shao, Fai Cheong Choo, Chetan Ganjihal, Xiangfa Guo, Padmanabha Venkatagiri. S and Manjunath Doddavenkatappa.

Having spent a couple of years in the Networking Department of I^2R, I am grateful to friendship and advice provided by the friends and collaborators whom I have gotten to know: Junxia Zhang, Xia Li, Kevin Zheng, Inn Inn Er, Choong Hock Mar, Jing Xie, Ricky Foo, Mingding Han, Winston Seah, Peng-Yong Kong, Wendong Xiao and Chen-Khong Tham.

In addition, I would like to take this opportunity to thank my friends, who

have been very understanding and supportive all this while - especially during conference paper deadlines. And to my family who has accompanied me through the years - thank you so much for all the encouragement and unconditional love.

Finally, this dissertation is dedicated to Zhongwen - who has constantly been my light in moments of darkness, and hope in times of despair.

Contents

Summary

The small wireless network devices in sensor and ad hoc networks can be deployed for a plethora of ubiquitous and collaborative applications, such as healthcare monitoring and tactical surveillance. However, these network elements are typically energy constrained as they have limited and/or irreplaceable battery supplies. This necessitates the design and development of energy efficient communication protocols in order to prolong the lifetimes of such networks.

In this dissertation, we first identify the caveats of existing networking protocols for energy constrained networks. Three novel algorithms, viz. A^2-MAC, IQAR and IQDEA, are then proposed to provide better energy efficiency for both periodic monitoring as well as event driven sensor applications.

A^2-MAC is an *A*daptive, *A*nycast *M*edium *A*ccess *C*ontrol protocol that effectively reduces energy expenditure in generic low-powered wireless sensor networks. It utilizes: (i) random wakeup schedules, such that each node can independently and randomly wakeup in each cycle without coordination and time synchronization; (ii) adaptive duty cycles based on network topology; and (iii) adaptive anycast forwarder selection, which allows each node to transmit to any member in its forwarding set. By allowing nodes to operate with different duty cycles and forwarding sets based on a given local delay performance objective and local network connectivity, A^2-MAC achieves better energy-delay tradeoffs and extends node lifetime substantially, while providing good end-to-end latency.

Upon the presence of Phenomena of Interest (PoI) in event driven sensor

networks, multiple sensors may be activated, leading to data implosion and redundancy. **IQAR** is an *I*nformation *Q*uality *A*ware *R*outing protocol that finds the least-cost routing tree that satisfies a given information quality (IQ) constraint when a PoI occurs. As the optimal least-cost routing solution is a variation of the classical NP-hard Steiner tree problem in graphs, IQAR uses: (i) topology-aware histogram-based aggregation structure that encapsulates the cost of including the IQ contribution of each activated node in a compact and efficient way; and (ii) greedy heuristic to approximate and prune a least-cost aggregation routing path.

Despite the existence of energy-delay tradeoffs, existing protocols tend to optimize only energy efficiency and overlook the significance of end-to-end delays. However, in mission critical applications such as intrusion detection and tsunami detection, faster detection of the PoI translates to earlier deployment of search-and-rescue operations and subsequently, significant reductions in casualties and infrastructural damages. **IQDEA** is an *I*nformation *Q*uality aware *D*elay *E*fficient *A*ggregation scheme that minimizes PoI detection delays and transmission costs in duty cycled networks while satisfying application-level IQ requirements. Through the use of: (i) IQ-awareness; (ii) novel aggregation latency function for each node; and (iii) selection of forwarding nodes based on instantaneous expected end-to-end delays, IQDEA achieves a good balance between energy efficiency and delay efficiency.

Performance evaluations of the proposed schemes show that they can achieve significant energy savings over existing protocols through the use of techniques such as adaptation to network conditions, anycast forwarding and information quality awareness. However, the design space for energy efficient communications remains very large, and continued research efforts are required to identify an integrated framework for the suite of these communication protocols.

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

Advancements in wireless networking and microelectromechanical systems (MEMS) technology have led to the proliferation of tiny computing and sensing devices that are often deployed in large numbers to perform collaborative tasks. A representative class of these networks is Wireless Sensor Networks (WSNs) [1] [2], which can be used for a multitude of applications - ranging from tactical or military surveillance, intruder detection, industrial automation, wildlife tracking, habitat monitoring, environmental monitoring, structural monitoring to healthcare monitoring. In these systems, characteristics of the physical environment (e.g. temperature, pressure, humidity and salinity) are sensed and transmitted via multihop links to a centralized fusion center (or sink) for processing and statistical analysis. As illustrated in Figure 1.1, such applications can generally be classified into two main categories, viz. periodic monitoring and event detection.

In periodic monitoring applications, data is collected from all the sensor nodes at regular intervals. The data is collected over a long period of time - in terms of weeks, months or even years - and is generally delay tolerant. Such data is then used to provide a statistical or analytical profiling of the terrain, environment or objects of interest. In the ZebraNet project [3], sensory data is collected

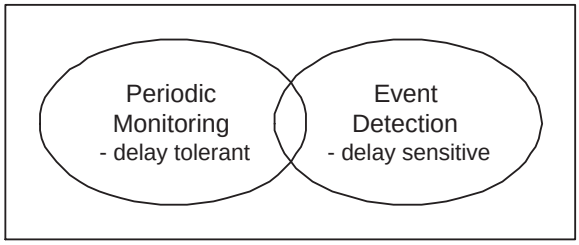


Figure 1.1: Classification of sensor network applications.

from tracking collars worn on animals of interest to provide an understanding of their migration patterns and inter-species interactions. On Great Duck Island off the coast of Maine [4] [5], sensors are deployed to monitor the habitat and nesting environment of seabirds, and provide live streaming data on the web. As compared to conventional instrumentations and methods of monitoring, the use of sensor networks for monitoring purposes has the advantage of providing fine-grained data at high resolutions with minimal invasion to natural habitats.

In event detection applications [6] [7], the primary objective is to detect a Phenomenon of Interest (PoI) when it occurs - such as an impending tsunami along the coastline [8], flood [9], forest fire [10] or an elderly person falling down at home [11]. Consequently, data collected for these applications is delay sensitive. The detection of a PoI can be achieved via naive methods (such as a sensor reading that is classified as an outlier in a statistical distribution), or via more sophisticated methodologies involving data aggregation and fusion. With the use of sensor networks for PoI detection, critical events can be reliably detected within pre-specified delay constraints, leading to the timely initiation of search and rescue operations.

Despite the apparent usefulness of wireless sensor networks, their successful deployments and operations face many challenges. The connectivity of wireless links [12] [13] is intermittent and temporal, and highly susceptible to environmental influences, which diminishes the predictability and reliability of packet transmissions. The limited radio range of sensor nodes and the large terrain of

deployment necessitate the use of multihop communication where intermediate relays are required to transmit data from each sensor source to the destination (fusion center) [1]. Node failures and topological changes may be prevalent if the network is deployed in harsh terrains such as mountainous or marine environments [14]. As sensor nodes are often densely deployed to provide data redundancy and maximize sensing coverage [15], there exists severe medium access contention during data transmissions. Even seemingly simple protocols such as flooding can exhibit complex behaviors which deteriorate network performance [16]. In particular, the severe energy constraints of sensor nodes have received much limelight in the research community [17], and is the focal point of the research in this dissertation.

1.2 The Case for Energy Efficient Communication

Sensor networks are expected to have a lifetime of several years; however, commonly used sensor platforms (such as Mica2, MicaZ, TelosB and Imote2 motes from the Crossbow family [18]) are powered by AA batteries, which severely limits their energy source. Furthermore, practical considerations such as inaccessible terrains and dense network deployments make it labor-intensive and unrealistic to physically replace each battery when it runs out. This leads to node failures and network partitions, which hinder inter-nodal communication, reduce data quality at the fusion center and deteriorate application-level performance.

In a sensor network, energy is expended through three main operations, viz. sensing, computation and communication [17]. During sensing, each node samples the physical environment at periodic intervals and converts the raw data into digital signals using Analog to Digital Converters (ADCs). Computational tasks include processing, data compression, as well as data aggregation and/or fusion. Inter-nodal communication, which take place in the form of packet transmissions and receptions, incur the bulk of total energy expenditure during the

lifetime of a node. Consequently, it is essential that each sensor node minimizes its energy consumption when communicating with its neighbor(s), in order to prolong overall network lifetime.

There exists a significant amount of work on energy efficient communication protocols for sensor networks in the literature [19] [20]. A key challenge in the design and development of such protocols is the ability to maximize energy savings and prolong network lifetime without excessively trading-off other performance metrics (such as delay and information quality of data at fusion center). This can be achieved only with the integration of energy-awareness at every stage of the network design and operation [17]. However, existing protocols typically consider only *one* aspect of the networking protocol stack, with noticeably concentrated efforts at the Medium Access Control (MAC) layer [21] [22] [23] [24] [25] [26] or network layer [27] [28].

We assert that many of these existing solutions have potential for improvements by incorporating the following techniques:

1. **Adaptation to local or prevailing network characteristics:** The network conditions and characteristics experienced by each node in a wireless sensor network vary with a wide range of factors - such as node location, local topology, traffic pattern and physical conditions. For instance, the underlying physical layer is subject to influences from the surrounding environment, leading to transient links that impede route establishment and maintenance. By ignoring the dynamics of the underlying link layer, a MAC protocol may repeatedly retransmit over the same intermittent link while a network routing protocol may select unreliable paths to the fusion center. This can lead to both excessive overheads and unnecessary energy consumption. Communication protocols that are adaptive to network characteristics are expected to react better to dynamic changes and hence provide better application-level performance.

2. **Exploitation of IQ-awareness:** The deployment of each sensor network is driven by an application-specific requirement on the information quality (IQ) of data that is collected at the fusion center. Existing literature frequently assumes that: (i) all sensory data is of equal importance; and (ii) all generated data is required at the fusion center. However, by leveraging the different IQ values provided by the sensory data, the system can intelligently acquire data with higher IQ and eliminate the need to collect data from all the nodes in the network. Energy expenditure can thus be minimized while satisfying application requirements.

Although energy efficient mechanisms have been proposed in the context of other classes of wireless networks - such as Wireless Local Area Networks (WLANs) [29] [30] and Mobile Ad Hoc Networks (MANETs) [31] [32] - these networks differ from WSNs in a myriad of ways. Unlike WLANs, sensor networks are decentralized, distributed and often have irreplaceable energy sources. While communications in WLANs and MANETs are often independent and point-to-point, data in sensor networks tends to be spatially and temporally correlated, and flows unidirectionally towards the fusion center in a convergecast fashion. Traffic in sensor networks is also sporadic in nature; it can either be triggered periodically (in monitoring applications) or event driven (in PoI detection applications). Hence, network solutions for the general classes of wireless networks are inadequate for sensor networks due to the unique characteristics of the latter.

1.3 Research Goals and Contributions

The main objective of our research work is to design and develop communication protocols for energy constrained networks to achieve energy efficiency while maintaining good energy-delay tradeoffs. We focus on wireless sensor networks as a class of energy constrained networks which are generally static, have little or no mobility, and have limited battery supplies. Ideally, these protocols should

prolong network lifetime, without overly compromising on other performance metrics that are of interest to the application. In this dissertation, we present three novel energy efficient communication protocols that not only address the caveats of existing protocols, but also achieve good tradeoffs for energy constrained networks: (i) **A^2-MAC** [33] - *A*daptive, *A*nycast MAC protocol; (ii) **IQAR** [34] - *I*nformation *Q*uality *A*ware *R*outing protocol; and (iii) **IQDEA** - *I*nformation *Q*uality aware *D*elay *E*fficient *A*ggregation scheme.

The MAC protocol is the key mechanism to enable communications between nodes in a network. A^2-MAC is an asynchronous and adaptive MAC protocol that utilizes: (i) random wakeup schedules, such that each node can independently and randomly select its wakeup schedule without coordination and time synchronization; (ii) adaptive duty cycles based on network topology; and (iii) adaptive anycast forwarder selection, which allows each node to transmit to any member in its forwarding set and effectively reduce expected sleep latency. By exploiting the redundancy from typical dense sensor network deployments, as well as combining random schedules and anycast mechanisms, nodes can operate with different duty-cycles and forwarding sets to reduce energy consumption, subject to a given delay constraint.

Despite the energy savings achieved through the use of a duty cycled MAC such as A^2-MAC, energy can still be expended unnecessarily due to data implosion and redundancy [35] arising from the activation of multiple sensors in event driven sensor networks. IQAR is an information quality aware routing protocol that aims to find a least cost (minimum energy) routing tree that satisfies a given IQ constraint within a delay bound. As the optimal least cost routing solution is a variation of the classical NP-hard Steiner tree problem in graphs, IQAR utilizes: (i) a topology-aware histogram-based aggregation structure that encapsulates the cost of including the IQ contribution of each activated node in a compact and efficient way; and (ii) a greedy heuristic to approximate and prune a least cost aggregation routing path.

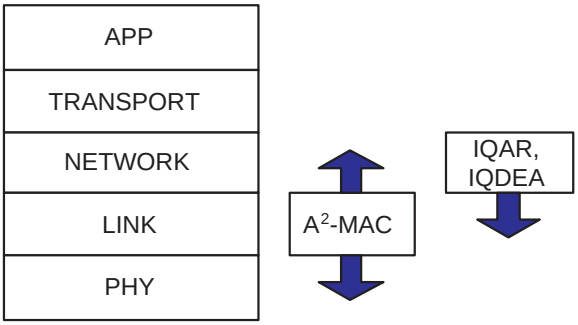


Figure 1.2: Cross-layer interactions between A^2-MAC, IQAR and IQDEA with the networking protocol stack.

In mission critical applications (such as intrusion detection or flood detection), PoI detection delay is a crucial performance metric as it determines how quickly search and rescue operations can be initiated in response to the PoI. Due to energy-delay tradeoffs that are inherent in data aggregation schemes [36], energy consumption is often minimized at the expense of longer detection delays. IQDEA is a data aggregation scheme that aims to minimize the event detection delay in a duty cycled network, without compromising on energy efficiency. A novel aggregation schedule is used to allow nodes to aggregate data efficiently while minimizing the PoI detection delay. Forwarding nodes are dynamically selected at each hop based on the instantaneous expected end-to-end delay and aggregated IQ at each neighbor. Through IQ-awareness, IQDEA terminates data acquisition as soon as sufficient data has been collected for reliable and accurate PoI detection. Hence, it is able to achieve good energy-delay tradeoffs while satisfying IQ requirements at the fusion center.

Figure 1.2 illustrates the cross-layer interactions between A^2-MAC, IQAR and IQDEA with the different layers in the networking protocol stack. Although A^2-MAC resides in the link layer, it utilizes forwarding set information from the network layer and feedback about the physical connectivity between links from the physical layer to adapt duty cycles and make a forwarding decision in real-time. Both IQAR and IQDEA reside in the network layer, and utilize

Table 1.1: Summary of Research Contributions

Protocol	**Description**
A^2-MAC	Adaptive, asynchronous MAC protocol that dynamically assigns a different duty cycle and forwarding set to each node based on its local topology, in order to minimize energy consumption subject to a delay constraint.
IQAR	IQ-aware routing protocol that builds a least-cost aggregation path in real-time to minimize energy consumption, subject to IQ and delay constraints.
IQDEA	IQ-aware data aggregation scheme that assigns aggregation latencies and selects forwarding nodes dynamically based on aggregated IQ and expected end-to-end delays, for the purpose of achieving good energy-delay tradeoffs.

instantaneous connectivity information from the bottom layers to make dynamic forwarding decisions. Through such loosely-coupled cross-layer interactions, A^2-MAC, IQAR and IQDEA are able improve overall network performance.

The contributions of this dissertation are summarized in Table 1.1. Although we focus on sensor networks as a representative class of energy constrained networks in this dissertation, the design philosophies are applicable to other generic networks with energy limitations.

1.4 Organization

The rest of this dissertation is organized as follows: Chapter 2 discusses background and related work on energy efficiency in wireless sensor networks. Protocol details and performance studies of A^2-MAC are described in Chapter 3. In Chapter 4, we present and evaluate IQAR, which constructs least-cost aggregation trees to achieve energy savings in real-time when phenomena of interest are detected in event driven sensor networks. In Chapter 5, we propose IQDEA, an IQ-aware data aggregation scheme that achieves a good balance between energy efficiency and delay efficiency while satisfying application-level IQ constraints. We conclude our work in Chapter 6 with directions for future research.

Chapter 2

Energy Efficiency in WSNs

While the development of energy efficient sensor network protocols is motivated by the need to extend 'network lifetime', the term *network lifetime* per se has taken on several definitions in the literature. In this chapter, we first discuss the various definitions of network lifetime and energy consumption characteristics in wireless sensor networks. We then present a survey of existing energy efficient communication protocols in the sensor network literature.

2.1 The Definition of Network Lifetime

Multiple definitions of network lifetime exist in the literature. It has been defined as time until the first node in the network dies [37] [38] [39] [40] [41], time until a percentage of the network dies [42], as well as time until the network does not satisfy application requirements [43] [44] [45]. In the following, we evaluate the accuracy and limitations of each of these definitions in capturing the essence of network lifetime.

The time until the first node of the network fails due to depletion of energy is useful in studies that aim to provide an even distribution of energy consumption and/or residual energy across all the sensors in the network. However, there is high spatial and data redundancy in typical sensor networks where nodes are

densely deployed in the monitored terrain. Nodes that are co-located within the same geographical region tend to monitor the same environment and/or detect the same phenomena of interest (PoI). A single node failure resulting from energy drain is unlikely to have negative effects on information quality of data at the fusion center or overall network performance. As such, defining network lifetime as the time until the first node failure inordinately underestimates the length of time during which the system is useful.

Conversely, defining the network lifetime as time until a certain percentage of the nodes dies may be an over-optimistic estimation of the length of time which the system is useful. In random node deployments, some nodes may form weakly connected components in the network. When nodes that serve as 'bridges' (v_4 and v_5 in Figure 2.1) between these weakly connected nodes and the rest of the network fail, partitions can ensue. In addition, nodes that are nearer to the fusion center (v_1, v_2 and v_3 in Figure 2.1) participate more frequently in data forwarding. Consequently, these nodes are likely to deplete their energy resources earlier than the rest of the nodes in the network. While the failure of these small subsets of nodes may not be sufficient to quantify the "percentage of nodes that die" before the network lifetime is reached, it can adversely impact the functionality, connectivity and spatial coverage of the network, as well as the quality of the data collected at the fusion center.

More recently, network lifetime has been defined to be the time duration before the network fails to satisfy its application or quality requirements - such as PDR (Packet Delivery Ratio), latency, throughput, connectivity, etc. Alfieri et al [43] defines network lifetime to be "time period from the time instant when the network starts functioning till the network runs satisfying its quality requirements". Suzuki et al [44] considers it to be "period in which the data arrival ratio is 100%", while Tang and Xu [45] annotates it as "time duration before it (the network) fails to carry out the mission due to insufficient number of alive sensor nodes". While these definitions appear to have more relevance to the application

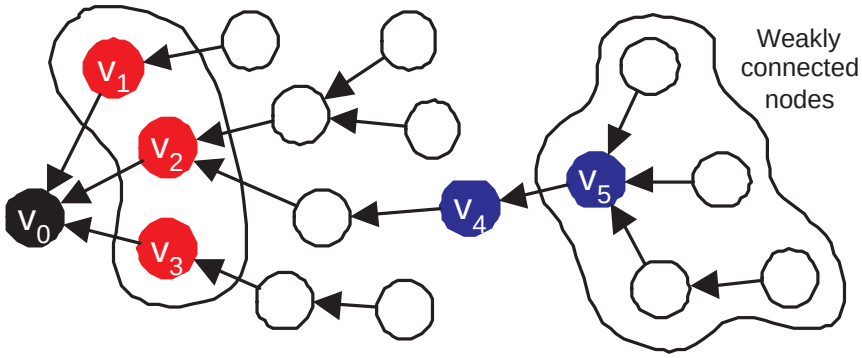


Figure 2.1: Nodes that are nearer to the fusion center (v_1,v_2,v_3) and nodes which act as bridges (v_4,v_5) for weakly connected nodes tend to fail earlier than the rest of the network.

scenario, care must be taken to define system requirements with sufficient comprehensiveness and completeness. For instance, an application requirement of 100% data arrival ratio is not particularly useful in real-time monitoring systems if the average end-to-end delay incurred to achieve this arrival ratio is excessively large.

Despite the many ambiguous definitions of *network lifetime* in current literature, it is unequivocal that the lifetime of a network is dependent on:

- energy expended in data transmission and reception;
- total energy expended in routing data from source to fusion center;
- statistical deviation of energy expended by nodes in the network; and
- amount of data required to meet application-specific performance (e.g. delivery ratio and detection accuracy).

As network lifetime is highly correlated with the energy expenditure by each node, it is important that energy efficient protocols are used in WSNs. The optimization of these protocols can be achieved only through a thorough understanding of energy consumption characteristics in each sensor node, as detailed in the next section.

Table 2.1: Current Draw of Different Motes (in mA)

	Mica2	MicaZ	Imote2
SLEEP	0.001	0.001	0.39
IDLE	-	0.426	31
RX	10.0	19.7	44.0
TX	11.0	11.0	44.0

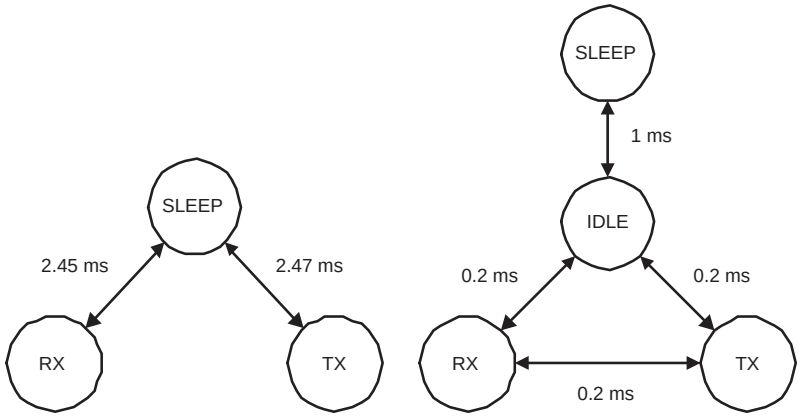
2.2 Energy Consumption in WSNs

According to [17], there are four subsystems in a canonical wireless sensor node: (i) computing subsystem (or microcontroller); (ii) communication subsystem; (iii) sensing subsystem; and (iv) power supply subsystem. In this dissertation, we focus on the communication subsystem - which has the primary objective of enabling wireless communications with other nodes - as it expends the most energy during the node lifetime.

The radio transceiver of a sensor mote is the core component in its communication subsystem, with many factors influencing its energy consumption. Some of these factors include modulation scheme, transmission range, data rate and operational mode. At any one time, the radio is in one of the following four operational modes - sleeping, idle, receiving and transmitting.

A node is in sleep (**SLEEP**) mode when its radio and voltage regulator are turned off; in this mode, it consumes the least energy and does not participate in any communication. In idle (**IDLE**) mode, the radio is turned on and the node is ready to receive incoming signals or transmit outgoing signals. A node in receiving (**RX**) mode is in the process of receiving a signal, which will subsequently be sent to the upper networking layers upon successful decoding. Finally, a node in transmitting (**TX**) mode is in the process of transmitting a signal to one or more of its neighbors.

Table 2.1 illustrates the typical values of the current drawn by some commonly used Crossbow motes when they are operating in the various modes,



(a) CC1000 Radio Model (b) CC2420 Radio Model

Figure 2.2: Simplified radio transition models.

while Figure 2.2 shows the simplified transition models of commonly used RF transceivers, such as CC1000 and CC2420 from Texas Instruments [46] which are used in Mica2 and MicaZ respectively. The timing on the edge of each transition indicates the approximate delays when switching from one mode to another.

Based on the radio models as summarized in Table 2.1 and Figure 2.2, the following key observations can be made:

- The SLEEP mode generally consumes the least amount of energy; hence nodes should be put to SLEEP instead of IDLE mode whenever possible.

- Mica2 does not have an explicit IDLE mode; instead, its radio transceiver is in RX mode when it is waiting to receive or transmit signals, which consumes higher energy as compared to the IDLE state in MicaZ. In addition, the cost of signal reception (RX) is higher than signal transmission (TX) in MicaZ; the converse is true in Mica2. Consequently, the design of the communication protocol should take into account the operating characteristics of the radio transceiver - such as the energy consumption for each mode and switching delays from one mode to another.

2.3 Energy Efficient Communication Protocols

Having studied the energy consumption characteristics of sensor motes, we now review some of the energy efficient communication protocols in current literature. Most of these existing solutions belong to one of the layers in the networking protocol stack, which comprises of the physical (**PHY**), data link (**LINK**), network (**NET**), transport (**TRANSPORT**) and application (**APP**) layers.

2.3.1 Energy Efficiency at the PHY Layer

The PHY layer is responsible for the transfer of sequences of bits between nodes sharing a wireless medium. At this layer, any transmission is subjected to interference and noise from the environment, leading to asymmetric links and frequent variations in signal quality. To aggravate the situation, the transmitted signal undergoes pathloss, fading and attenuation, which are all dependent on inter-nodal distance and the physical environment. Nevertheless, Shih et al [47] advocates the use of PHY layer approaches in the design of energy efficient protocols. Some of these schemes include advanced radio frequency circuits, modulation and channel coding schemes, as well as power or topology control.

In topology control, the transmission power of each node is dynamically adjusted to minimize energy consumption while maintaining network connectivity. Santi [48] provides a taxonomy of topology control techniques, which can broadly be classified as: (i) homogenous, where all nodes have the same transmission power; or (ii) non-homogenous, where nodes may have different transmission powers. However, Burkhart et al [49] shows that majority of these algorithms minimize energy consumption at the cost of increased interference, which may severely deteriorate network performance. Furthermore, many of these algorithms are centralized in nature, assume the knowledge of node locations, and incur significant overheads during the exchange of neighboring information.

2.3.2 Energy Efficiency at the LINK Layer

The LINK layer is responsible for the establishment of stable links over the unreliable wireless medium; this is usually done in the form of ARQ (Automatic Repeat reQuest) and FEC (Forward Error Correction) techniques. The Medium Access Control (MAC) layer is a sub-layer of the LINK layer that arbitrates access to the shared wireless channel among nodes in the network. Although MAC schemes can be classified as contention-free, contention-based or hybrid, sensor MAC protocols are typically contention-based due to the absence of a centralized controller that allocates resources to nodes in a multihop network.

In the pioneering work on sensor MAC protocols, Ye et al [50] identifies the main sources of energy consumption in any contention-based MAC protocol as: (i) collision; (ii) overhearing; (iii) control packet overhead; and (iv) idle listening. Collisions occur when nodes attempt to send packets over the shared wireless medium concurrently, leading to packet corruptions and retransmissions. Due to the broadcast nature of the channel, unicast packets may be overheard by neighboring nodes that are not the intended destinations. Control packets that are used for synchronization and network management compete with data packets for channel bandwidth. In sensor networks without energy awareness, nodes expend most of their energy in idle listening due to the sporadic nature of data traffic. Consequently, subsequent works on sensor MAC protocols always incorporate some form of wakeup scheduling such that nodes do not remain awake throughout the entire network lifetime but wakeup at intervals for communication and to check for channel activity.

Wakeup mechanisms can be broadly classified as: (i) on-demand; (ii) synchronous; and (iii) asynchronous. Sensor MAC protocols that make use of on-demand wakeup mechanisms [51] require out-of-band signaling (using a low power radio) in order to wake up nodes in time for data reception. At least two radios are required in these schemes - a low-powered radio that is constantly

awake to sense for any channel activity, and a high-powered radio which is awakened on-demand by the former whenever any activity is detected. However, complex algorithms are required to handle the differences in communication ranges of low-powered and high-powered radios [52].

In synchronous wakeup (or scheduled rendezvous) schemes [50] [53] [54] [55] [56], nodes wakeup during the same designated time slots to communicate. This effectively reduces idle listening and achieves low power consumption, albeit at the expense of long latency. Furthermore, tight time synchronization and pre-negotiation of schedules are necessary, which incur high overheads.

In asynchronous wakeup schemes [57] [58] [59], schedules of senders and receivers are decoupled, thereby removing the need for synchronization. Using a technique commonly known as LPL (Low Power Listening), nodes wake up periodically to check for channel activity. A node remains awake if channel activity is detected, and resumes sleeping otherwise. Extended preambles are required for the correct detection of channel activity, which increases delay and energy consumption.

2.3.3 Energy Efficiency at the NET Layer

At the NET layer, paths from the sensor sources to the fusion center are established and maintained by the routing protocol. Due to the scale of the network and limited transmission ranges of sensor nodes, communication typically takes place through multiple hops, in a distributed manner.

Typical routing protocols are based on shortest path algorithms that optimize performance metrics such as throughput and delay; however, they consider neither energy efficiency nor information quality. In contrast, energy efficient routing protocols aim to achieve one or both of the following goals while routing a packet from the source to the fusion center: (i) minimizing total energy consumption; and/or (ii) maximizing distribution of energy consumption such that time until the first node depletes its energy is prolonged.

In minimum energy routing schemes [60] [61] [62] [63], the edge of each node is associated with the (energy) cost of transmitting across that particular link. The routing algorithm will then select a route such that the sum of all the energy costs along that path is the lowest among all other possible paths. As transmission power is highly correlated with distance, routes with the smallest energy consumption are usually shortest-distance or smallest-hop paths. Although this can effectively reduce the overall energy consumption, it may lead to network partitions when nodes along paths that offer the least energy consumption are frequently used to forward data packets, causing their early depletion.

To minimize network partitions while reducing energy consumption, routing protocols that place emphasis on load or energy distribution have been proposed [37] [40] [41]. Instead of selecting routes that maximize energy savings, these routing protocols avoid routes through nodes with very low residual energy. This prolongs the time before any node along a path depletes all its energy and allows the network to degrade gracefully; however, this may lead to the establishment of sub-optimal paths with poor network performance.

Multipath routing [64] [65] [66] has also been widely considered as a technique to achieve load balancing, alleviate congestion, as well as to distribute energy consumption more evenly throughout the network. These routing protocols establish multiple routes throughout the network; packets are then routed through the paths in a round-robin or probabilistic manner.

2.3.4 Energy Efficiency at the TRANSPORT Layer

The need for a transport layer protocol to provide reliable data delivery in sensor networks is discussed in [67], whereby the authors suggest that although most sensor network applications are typically loss tolerant, messages that are initiated from the fusion center to the sensor nodes require guaranteed packet delivery. PSFQ (Pump Slowly, Fetch Quickly) is proposed as a reliable transport layer protocol for sink-to-source communications in wireless sensor networks.

Although PSFQ can provide high error tolerance, low communication overhead and support for loose delay bounds, it does not address energy constraints and packet losses caused by congestions.

ESRT (Event-to-Sink Reliable Transport) is subsequently proposed by Akan and Akyildiz [68], which aims to "achieve reliable event detection with *minimum energy expenditure* and congestion resolution". ESRT leverages temporal correlations in sensory data to ensure that event features at the fusion center do not exceed a particular distortion bound. It minimizes energy consumption by reducing the reporting frequency of sensor nodes while maintaining an acceptable level of data reliability.

aDapTN [69] aims to achieve energy efficiency by reducing the time spent on idle listening. aDapTN is based on the Delay Tolerant Network (DTN) architecture and is suitable for both intermittently connected networks and networks with long propagation delays. As part of a cross-layered design, it integrates a store-and-forward transport approach with an asynchronous wakeup scheme. Whenever a neighboring node along the routing path is asleep, aDapTN caches the message at the intermediate node until connection is resumed. The authors claim that this can achieve packet delivery reliability and reduce energy wastage caused by idle listening.

2.3.5 Other Energy Efficient Strategies

Data Aggregation and/or Fusion

High communication cost and data redundancy in energy-constrained sensor networks necessitate the use of in-network processing [35] [70] [71] to aggregate spatio-temporally correlated data for the primary purpose of reducing energy expenditure. Existing work on data aggregation can be classified as structured or structureless approaches.

The energy-optimal data aggregation structure for a known set of sensor

sources is the Steiner Minimum Tree (SMT) [72]. As construction of an optimal SMT is NP-hard and incurs significant overhead in large scale multihop networks, the Minimum Spanning Tree (MST) is often used as an approximation. Several heuristics that approximate SMT to achieve energy efficiency have also been proposed [73] [74] [75] [76] [77] [78]. Furthermore, there is significant work on delay-optimal scheduling algorithms for given aggregation structures in the literature [79] [80] [81] [82] [83] [84] [85]. Some cluster-based aggregation schemes [86] [87] [88] have also been proposed, whereby each cluster head collects data from multiple nodes within its cluster before forwarding the aggregated data to the fusion center directly. However, these cluster-based aggregation schemes are not popular as: (i) high transmission power levels are required to transfer data from each cluster head to the fusion center in a single hop; and (ii) excessive message overhead is incurred during periodic cluster head elections.

In many of these structured aggregation schemes, the aggregation latency at each node is typically staggered to allow data from child nodes to be transmitted to their corresponding parent nodes, so that the latter can aggregate the data together before forwarding it towards the fusion center. Although these schemes tend to minimize energy consumption due to ample aggregation opportunities, they generally incur high PoI detection delays. Furthermore, these structured protocols work on the premises that: (i) traffic pattern is invariant (e.g. in periodic monitoring applications); and (ii) construction and maintenance of a fixed data aggregation structure incur low overhead. Consequently, they are unsuitable for delay-critical event-driven applications such as intrusion detection systems [89] or bioterrorism detection systems [90] where sensor sources are not known *a priori*.

In structureless aggregation [91] [92], data aggregation takes place opportunistically, only when data from multiple sources arrive at the same time at a particular node. Due to this inefficiency in data aggregation, these schemes incur high energy consumption and do not scale well. However, as no additional

waiting latency is incurred due to aggregation, these schemes can achieve short PoI detection delays. Although semi-structured approaches [93] [94] [95] have been proposed to balance the tradeoff between aggregation efficiency and overheads incurred to maintain an aggregation structure, they do not exploit the information quality content of sensory data to improve energy efficiency.

Multiple Fusion Centers

The primary role of the fusion center in a wireless sensor network is to acquire data from sensor sources in the network. It is assumed to be a computing device with higher capabilities than the rest of the network elements - it may have a wired connection to other infrastructured networks such as the Internet, as well as possess untethered power supply, processing abilities and unlimited storage. Consequently, fusion centers are considered to be expensive devices that should be deployed sparingly in a sensor network.

Existing works frequently assume the presence of only *one* fusion center in a sensor network, placed either at the center or boundary of the monitored terrain. However, the location of the fusion center is associated with several issues:

1. Long routing paths with large hop counts are required to reach a fusion center that is placed far away from the sensor nodes. This may result in frequent packet losses and long end-to-end delays.

2. The funneling effect [96] is a culmination of the many-to-one traffic pattern in sensor networks, where multiple sensor sources transmit sensed data via multiple hops to a single fusion center. It can lead to excessive congestion, packet losses and energy wastage.

As such, the use of strategically located *multiple* sinks has been proposed as a means of traffic redirection, load balancing, path length reduction and energy reduction in sensor networks [44] [96] [97] [98] [99]. However, it may be unrealistic

to deploy multiple sinks at optimally-computed locations due to the hostility of the physical terrain and unavailability of *a priori* node locations.

2.3.6 Energy Efficiency in Other Wireless Networks

Besides wireless sensor networks, there exists a plethora of work on energy efficient protocols for other types of wireless networks[1], such as Wireless Personal Area Networks (WPANs) and Mobile Ad Hoc Networks (MANETs). In the following, we outline some of the existing protocols that focus on energy efficiency in these networks and discuss their applicability in WSNs.

Wireless Personal Area Networks (WPANs)

WPANs are made up of pervasive, mobile computing devices such as smartphones, laptops and Personal Digital Assistants (PDAs) that communicate via wireless technologies such as Bluetooth [100]. As like sensor nodes, these devices are small in size and battery-operated.

To minimize energy consumption during periods of low activity [101], three modes of operation are introduced in the Bluetooth technology, viz. hold, park and sniff. [102] evaluates each of these modes and show that the sniff mode has the smallest response time while park mode incurs the least energy consumption. [103] proposes ASP, an adaptive energy efficient polling algorithm for bluetooth piconets in which sources send short data packets at constant rates.

As the operating ranges of these networks are expected to be small (in the range of 10 to 20 meters), these energy aware schemes are typically designed for single hop communication. For example, the conventional Bluetooth architecture allows a master node to communicate with up to seven slave nodes in the same piconet. Although multiple piconets can be interconnected to form scatternets, energy efficient approaches [104] [105] [106] for this architecture tend to focus

[1]We focus only on wireless networks as wired networks are often implicitly assumed to be connected to untethered energy supplies.

on minimizing the energy required to: (i) form and maintain the scatternets; and/or (ii) find routes between two nodes in the scatternet.

Mobile Ad Hoc Networks (MANETs)

MANETs are wireless networks that offer multi-hop connectivity between self-organizing and self-configuring mobile hosts. As like in WSNs, each node in a MANET functions as both a host as well as a router to forward packets to other nodes. Many of the energy efficient works on MANETs in the literature focus on the minimization of energy consumption during route discovery and maintenance [107] [108] [109] and load balancing [110]. However, these schemes cannot be directly applied to WSNs due to the differences in network characteristics. While nodes in MANETs are assumed to have mobility, nodes in WSNs are assumed to be relatively static. Link connectivity in MANETs is assumed to be intermittent due to node mobility; in WSNs, link breakages occur due to duty cycling or node failures resulting from energy drain.

2.3.7 Summary

In this chapter, we study the issue of energy efficiency in wireless sensor networks. The discussion on network lifetime further accentuates the need for energy efficient protocols, and establishes that network lifetime should be defined based on a comprehensive set of application specific requirements. The main sources of energy consumption in a wireless sensor node are identified, which provides a better understanding of how effective energy efficient protocols should be designed. We have also surveyed the existing energy efficient communication protocols and provided a summary in Table 2.2. Based on our study, we observe that the design of many of these protocols can be improved upon by: (i) leveraging sensor network characteristics such as dense node deployments and node redundancy; (ii) incorporating information quality awareness; (iii) incorporating cross-layer interactions; and (iv) adapting to network characteristics. In the next three

chapters, we demonstrate that by taking these factors into consideration, the energy efficiency of communication protocols can be further improved upon.

Table 2.2: Techniques to Achieve Energy Efficiency in Communication Protocols

Layer	Technique	Description
PHY	Topology Control [48] [49]	Adjustment of transmission power to minimize energy consumption while maintaining network connectivity.
LINK	Duty Cycling [50] [111] [51] [50] [53] [54] [55] [56] [57] [58] [59]	Due to sporadic traffic, nodes sleep periodically to minimize idle listening.
	Overhearing Avoidance [50] [111] [51] [50] [53] [54] [55] [56] [57] [58] [59]	Nodes sleep upon knowledge that channel is going to be used for other transmissions.
	Collision Avoidance [50] [111] [50] [53] [54] [55] [56] [57] [58] [59]	Nodes sleep or refrain from sending packets when channel is busy.
NET	Minimum Energy [60] [61] [62] [63]	Selection of routes based on minimum energy cost along the entire path.
	Max-min [37] [40] [41]	Selection of routes based on the maximum remaining energy of nodes along the path, or avoidance of nodes with very minimal remaining energy.
	Multipath [64] [65] [66]	Selection of multiple routes towards fusion center for load balancing.
TRANSPORT	Minimization of sensor reporting frequency [68]	Leverages temporal correlation to reduce reporting frequency of sensor nodes without compromising on data reliability.
	Store-and-forward with wakeup scheduling [69]	Message is cached and forwarded later, upon temporal connectivity between node pairs.
APP	-	-
Others	Data Aggregation and/or Fusion [35] [70] [71] [79] [80] [81] [82] [83] [84] [85] [86] [87] [88] [91] [92] [93] [94]	Reduces traffic volume by leveraging temporal and/or spatial correlation to aggregate/fuse data together.
	Multiple sinks [44] [96] [97] [98] [99]	Reduces path length and provides load balancing for balanced distribution of energy dissipation.

Chapter 3

Adaptive, Anycast Medium Access Control

In this chapter, we detail the motivation, protocol design and performance studies of A^2-MAC [33], an adaptive, anycast Medium Access Control (MAC) protocol for Wireless Sensor Networks.

3.1 The Case for Duty Cycling

Due to the sporadic nature of sensory traffic, sensor nodes are prone to idle listening - which has been identified as one of the primary sources of unnecessary energy expenditure in WSNs [50]. By incorporating duty cycling into MAC operations, nodes need not monitor the channel continuously for communications. Each node remains in low-power sleep mode most of the time, and wakes up periodically to sense for any channel activities.

Performance studies [21] [50] [53] [54] [55] [56] [57] [58] [59] show that while wakeup schedules are effective in reducing energy consumption of sensor networks due to the sporadic characteristics of sensor traffic, the delay incurred by waiting for the next-hop forwarding node to be awake, viz. *sleep latency*, can be quite large. For example, a 1% duty cycle can potentially reduce the energy

consumption of a network by 99% when no traffic is being generated. However, the expected per-hop sleep latency of a packet is 50% of the cycle period, which can be up to a few seconds or more.

The wakeup schedule is a key component in the design of a duty cycled MAC to reduce energy consumption. Synchronous schemes such as S-MAC [21] [50] [53], T-MAC [54], D-MAC [55] and R-MAC [56] require synchronization among nodes, which can be complex and expensive especially in large multihop networks with clock drifts, low duty cycles and transient link qualities. Reduction in sleep latency is thus achieved at the expense of substantial control overhead. Asynchronous schemes such as B-MAC [57], X-MAC [58] and C-MAC [59] rely on preambles to coordinate access to the channel and do not require synchronization. Such schemes are energy efficient for low data traffic but incur long sleep latencies. Thus, there exists an obvious tradeoff between energy savings and latency incurred using wakeup scheduling.

3.2 The Case for Adaptive and Anycast Paradigms

A key difference between many existing duty cycled MAC protocols and A^2-MAC is that the latter employs *adaptive* and *anycast* paradigms in its protocol design. In this section, we justify why these two methodologies are essential in an energy efficient MAC protocol for wireless sensor networks.

In large-scale sensor networks, it is impractical to place each node in a strategic, pre-planned location. Instead, sensor nodes are usually randomly distributed in the monitored terrain with sufficiently high density to ensure coverage and connectivity; hence, the distribution of the nodes in the network is likely to be non-uniform, with varying local connectivity and density. However, existing MAC protocols tend to assign the same duty cycle to each node in the network, without taking into account the local network topology or exploiting the redundancy resulting from the denseness of the deployment. In such scenarios, each

node is assigned the same (high) duty cycle to meet the delay constraints, and all the nodes in the network will fail from energy drain at about the same time, resulting in premature termination of the usefulness of the network. A²-MAC avoids this situation by: (i) adapting the duty cycles of each node based on its local topology, which allows the network to fail gracefully over time; and (ii) exploiting node redundancies to prolong network lifetime.

The wireless medium is also characterized by intermittent and temporal connectivity [12] [13], leading to unreliable communication links that deteriorate network performance. Gu and He [112] studies the impact of these unreliable links on data forwarding in duty cycled networks and asserts that with low duty cycles, link quality measurements performed previously are likely to be outdated. Consequently, the use of a fixed forwarder or fixed forwarding set can be detrimental in a dynamically changing network, as a node may be forwarding data to a neighbor that has permanently failed, or repetitively retransmitting data across a link that has failed intermittently.

A²-MAC alleviates this situation through the utilization of an anycast mechanism, which exploits path diversity by allowing the transmitting node to send packets to any one member in its forwarding set [59] [113] [114] [115]. To prevent routing loops, the forwarding set typically includes only nodes with positive 'progress' towards the destination; the next-hop node is then chosen dynamically based on prevailing link conditions. In [115], MAC and routing functions are combined and utilized together with end-to-end connectivity information to optimize sleep-wake scheduling as well as maximize time to first node failure. However, the same duty cycle is used for all the nodes in the network.

In this chapter, we provide details of *A²-MAC*, an asynchronous *A*daptive, *A*nycast *MAC* protocol for low-powered sensor networks. It utilizes: (i) a *random wakeup schedule*, such that each node can independently and randomly select its wakeup schedule without coordination and time synchronization; (ii) *adaptive duty cycles* based on network topology; and (iii) *adaptive anycast forwarder*

selection, which allows each node to transmit to any member to its forwarding set and effectively reducing expected sleep latency.

There are two key adaptations in A^2-MAC: (i) each node adaptively varies its duty cycle and set of forwarding nodes such that energy consumption is locally minimized for a delay performance objective; and (ii) nodes cooperatively reduce the duty cycles required of their forwarding nodes, depending on local network connectivity. By exploiting the redundancy of dense network deployments as well as combining random schedules and anycast mechanisms, nodes can operate with different duty cycles and forwarding sets to reduce energy consumption. We compare A^2-MAC with X-MAC [58] and the optimal protocol in [115] (hereafter referred to as opt-MAC for brevity) whereby all nodes use the same duty cycle. Our performance evaluation shows that A^2-MAC can achieve better energy-latency trade-offs and extend node lifetime substantially while providing good end-to-end latency.

3.3 Protocol Details of A^2-MAC

3.3.1 System Model

The network is defined as a graph $G = \{V, E\}$ where V denotes the set of n sensor nodes and E denotes the set of edges. The wakeup schedule of A^2-MAC is based on an asynchronous slot model, which eliminates the need for costly time synchronization among different nodes in the network. Instead, each node only needs to maintain local synchronization such that slots within a node are of the same length during the network lifetime. The schedule in each cycle is divided into: (i) active (listening) slots, in which nodes wakeup and monitor the channel for activities (analogous to Low Power Listening in asynchronous MAC schemes); and (ii) inactive (sleep) slots, in which nodes switch to low-powered sleep mode by default. In each cycle, n_s is the total number of slots (chosen to achieve a sleep latency constraint), and τ is the duration of each slot in the

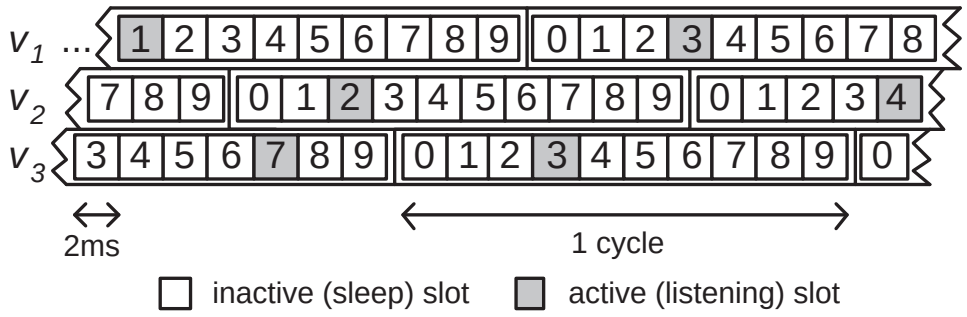


Figure 3.1: (10, 1, 2ms) random wakeup function with 3 unsynchronized nodes.

duty cycle such that cycle length is $n_s\tau$. Figure 3.1 illustrates a duty cycle with $n_s = 10$ slots and slot duration $\tau = 2$ms. Due to the asynchronous nature of the MAC model, slots of each node may be unsynchronized with other nodes.

3.3.2 Basic Mechanism

Each node v_i follows a random wakeup scheduling function represented as a (n_s, α_i, τ) design, where $\alpha_i \leq n_s$ is the number of *active* listening slots per cycle. At the beginning of each cycle, v_i selects α_i out of n_s slots to be active in, such that its active/awake probability in any slot is $\frac{\alpha_i}{n_s}$. The locations of these α_i slots are selected randomly and independently of other nodes to: (i) eliminate coordination and synchronization overheads; and (ii) provide ease of adaptation of duty cycles. However, the choice of α_i is dependent on duty cycle requirements of each neighbor v_k that uses v_i to forward packets to the fusion center. We refer to v_k as an **upstream** node of v_i; in the same manner, v_i is considered a **downstream** node of v_k.

Let α_{ki} be the duty cycle requirement **of** v_i by v_k; this refers to the duty cycle that v_k requires its downstream node v_i to have, in order to satisfy delay constraints[1]. The duty cycle of v_i is then given by the maximum of all the duty

[1]The duty cycle requirement α_{ki} by v_k is elaborated in further detail in subsequent sections.

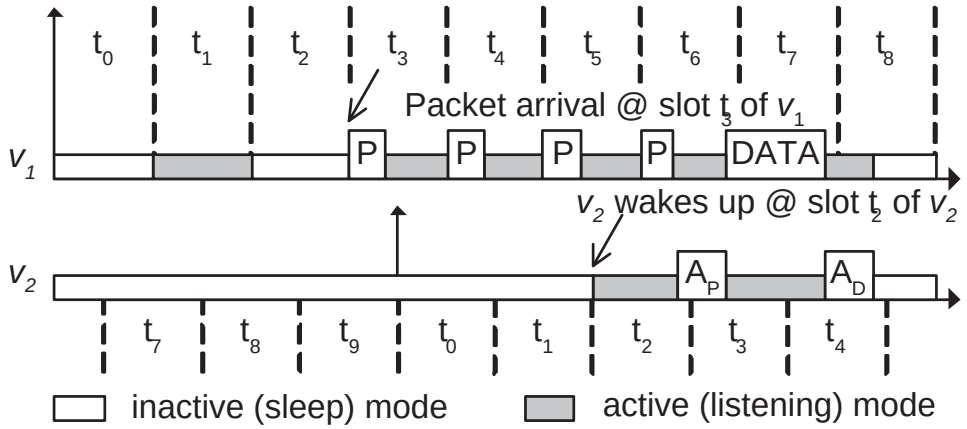


Figure 3.2: Data transfer in A^2-MAC between 2 unsynchronized nodes.

cycle requirements **from** its upstream nodes, such that:

$$\alpha_i = \max_{v_k \in V_i^u} \alpha_{ki}, \tag{3.1}$$

where V_i^u is the set of *upstream* nodes of v_i.

In the (10, 1, 2ms) random wakeup function of Figure 3.1 (with $\alpha_1 = \alpha_2 = \alpha_3 = 1$), nodes v_1, v_2 and v_3 each selects 1 active slot out of $n_s = 10$ available slots in a cycle. Note that it is also possible for each node to select a different α_i value.

Compared to synchronous schedules, the number of *active* one-hop neighbors during an arbitrary time slot in A^2-MAC is reduced, which effectively minimizes collisions and reduces overhearing. However, the default active slots of each node are unlikely to overlap, particularly when duty cycle is low. This can result in the 'lonely node' problem [116] that is inherent in low duty cycled asynchronous MAC protocols, in which nodes are unable to find any neighbors to communicate with upon waking up. A^2-MAC uses a probing mechanism to guarantee communication between the transmitter and its forwarder (if one exists) within a single cycle period.

In Figure 3.2, v_1 wakes up at (its) slot t_1 according to its wakeup function in Figure 3.1 and resumes sleeping at t_2. Upon a packet arrival at t_3, v_1 switches to

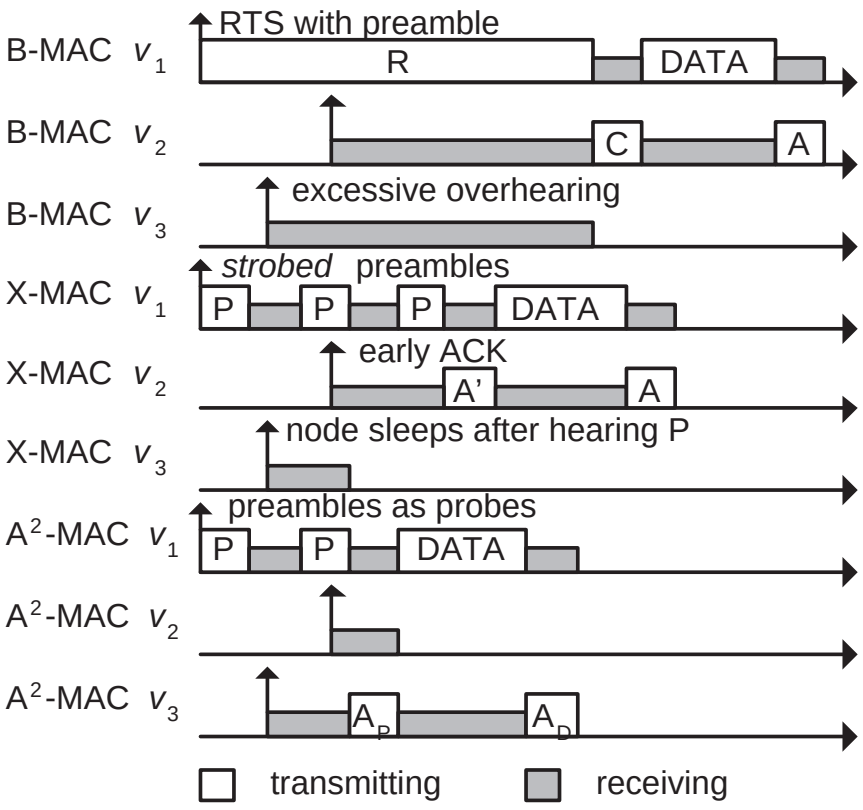


Figure 3.3: Protocol details of B-MAC, X-MAC and A^2-MAC.

active (listening) mode and continuously probes its neighbors using short preambles P in every subsequent slot. Each preamble transmission lasts for less than the duration of a slot length. Probing terminates when v_1 receives a preamble acknowledgement A_P at t_6 from a forwarder v_2 that is awake. Transmission completes when v_1 transmits data to v_2 during t_7 and receives a corresponding data acknowledgement A_D in t_8. Note that in order for v_2 to detect that the channel is busy and subsequently receive a preamble P from v_1, it has to be awake for at least one slot length τ each time it is scheduled to wakeup. If v_2 senses that the carrier is not idle in its current active slot, it will stay awake in the next consecutive time slot. This allows it to receive any potential packets that may have been transmitted but not yet arrived, as is often the case with nodes that have unsynchronized schedules.

The probing for active neighbors does not incur additional delays or overheads as compared to existing asynchronous MAC protocols, as all such protocols *have* to transmit preambles *up to* duration of a cycle period to guarantee communication between nodes. Figure 3.3 illustrates behaviors of B-MAC [57], X-MAC [58] and A^2-MAC upon a packet arrival at time t_A at v_1. Due to the unicast nature of B-MAC and X-MAC, v_3 cannot forward packets for v_1 even though it wakes up before v_2, as v_2 is the designated forwarder for v_1. In contrast, A^2-MAC achieves shorter delays and incurs less overheads using anycast, as it allows any node in its forwarding set to forward data to the fusion center. Note that in Figure 3.3, v_2 goes back to sleep upon overhearing the acknowledgement A_P from v_3. In the case that v_2 is not within the range of v_3, it will go back to sleep after v_1 has finished its data transmission.

When collisions of A_Ps occur due to the presence of multiple awake forwarders that detect P and transmit their A_Ps at the same time, each forwarder backoffs for a randomly chosen interval before attempting to retransmit its A_P. However, as the duty cycle and traffic load of the network are expected to be very low, the corresponding probability of such collisions is low. When there is only one forwarding node, A^2-MAC behaves similarly to X-MAC.

3.3.3 Combination of Anycast with Random Schedules

Besides path diversity and multipath routing [59] [117] [118], the anycast mechanism used in A^2-MAC can provide other advantages, such as: (i) robustness to intermittent link connectivity; and (ii) latency reduction in a duty cycled MAC.

Robustness to Intermittent Link Connectivity

The transient characteristics of the physical layer leads to intermittent link connectivity. Typical MAC protocols attempt multiple retransmissions across the same temporally-broken link before a link failure is ascertained and an alternative routing path is utilized. With anycast, a node can dynamically select its

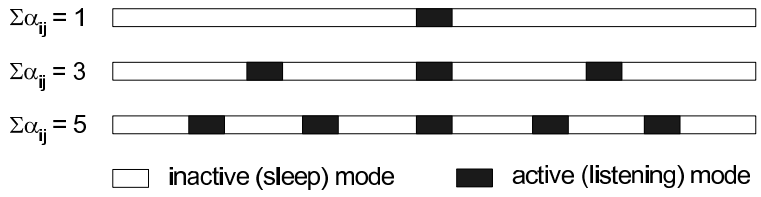


Figure 3.4: Uniform random distribution of active slots with varying $\sum_{v_j \in F_i} \alpha_{ij}$.

forwarder based on prevailing link conditions and provide robustness to intermittent connectivity. This achieves load balancing, alleviates effects of temporary link failures, reduces delays and reduces retransmission overheads.

Reduction in Latency

In contrast to unicast schemes where a transmitter has to wait for a particular forwarding node to be awake, A^2-MAC enables the transmitter v_i to send packets to any node in its forwarding set F_i as soon as one of them is awake. Since the active slots in a cycle are randomly chosen, they can be viewed as being uniformly distributed among the remaining inactive slots, as illustrated in Figure 3.4.

We define **average sleep latency** T_i to be the time (measured in slots) incurred by node v_i before any one of its forwarders $v_j \in F_i$ is awake. Hence, T_i is given by:

$$T_i = \frac{n_s - \sum_{v_j \in F_i} \alpha_{ij}}{\sum_{v_j \in F_i} \alpha_{ij} + 1}, \tag{3.2}$$

where α_{ij} is the duty cycle requirement **of** each forwarder $v_j \in F_i$ by v_i.

In typical sensor networks, which have low duty cycles, the total number of active slots required by a node is generally very small in comparison to the number of slots in each cycle (i.e. $n_s = 100 >> \sum_{v_j \in F_i} \alpha_{ij} = 1$). Consequently, the sleep latency T_i in Equation 3.2 can be approximated by:

$$T_i \approx \frac{n_s}{\sum_{v_j \in F_i} \alpha_{ij} + 1}. \tag{3.3}$$

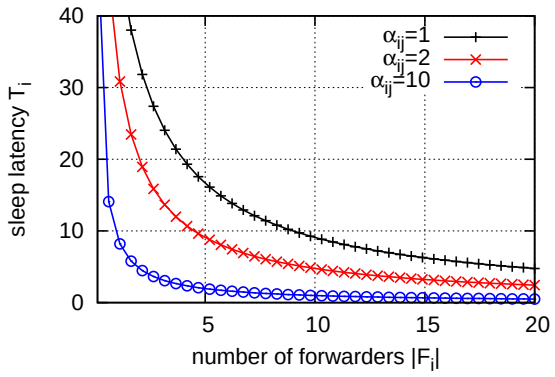


Figure 3.5: Sleep latency T_i with varying values of α_{ij} and $|F_i|$ when $n_s = 100$.

With a larger number of forwarders in F_i, the sleep latency of node v_i can be reduced. As illustrated in Figure 3.5, sleep latency T_i (plotted using Equation 3.3) decreases with increasing number of forwarders $|F_i|$ and/or average duty cycle requirements α_{ij} (assumed to be same for all the forwarders in F_i.

3.3.4 Interaction with Routing Protocol

A^2-MAC is inter-operable with any routing protocol that provides: (i) a set of candidate forwarding nodes; and (2) a metric that indicates the *progress* made by each forwarder. Examples of such metrics include hopcount to destination, geographical distance [118] and ETX [119]. In this chapter, we use the Maximum Forward Progress (MFP) [120] routing metric, which forwards packets based on geographical locations. Each node is assumed to have locations of itself and the fusion center, which can be obtained via GPS or existing localization schemes. Only neighbors with positive progress (closer to the fusion center than the transmitter) are considered to be eligible forwarders. As no state information is required in MFP, this allows us to study the performance of A^2-MAC without having to take into account routing overheads.

3.4 Adaptation in A^2-MAC

The primary objective of A^2-MAC is to reduce the energy consumption of nodes through duty cycling, in order to extend network connectivity and coverage, subject to a desired delay constraint $d_{\max}$. In this section, we describe the two key adaptation components of A^2-MAC, viz. forwarder selection and duty cycle selection, that help to achieve this objective.

The **candidate set** $\aleph_i$ of an arbitrary node v_i is the set of one-hop neighbors with *positive* progress towards the fusion center (destination). Using MFP as the routing protocol, neighbors with positive progress comprise the nodes that are closer to the fusion center than v_i is to the fusion center. $\aleph_i$ can be learnt via a simple neighbor discovery scheme during network initialization. For each candidate node $v_j \in \aleph_i$, p_{ij} denotes the progress made by v_i when it transmits data to the fusion center via v_j. $\alpha_{ij} \in (0, n_s]$ denotes the duty cycle required of v_j by v_i (in a cycle with n_s slots). The **forwarding set** $F_i \subseteq \aleph_i$ is the set of neighbors within the candidate set that are selected to forward packets from v_i to the fusion center.

The average (expected) per-hop progress P_i made by v_i is defined as:

$$\begin{aligned} P_i &= \sum_{v_j \in F_i} p_{ij} \cdot \frac{\alpha_{ij}}{\sum_{v_j \in F_i} \alpha_{ij}} \\ &= \frac{\sum_{v_j \in F_i} (p_{ij} \cdot \alpha_{ij})}{\sum_{v_j \in F_i} \alpha_{ij}}, \end{aligned} \tag{3.4}$$

and the corresponding average per-hop rate of progress S_i of v_i is given by:

$$S_i = \frac{P_i}{T_i}, \tag{3.5}$$

where T_i is the sleep latency in Equation 3.2. Generally, the inclusion of more forwarders decreases T_i; however, inclusion of forwarders with small progress (p_{ij}) decreases the average progress P_i and rate of progress S_i.

We denote the delay constraint (which is specified as an application requirement) as $d_{\max}$; thus, any data that satisfies the delay constraint should incur an end-to-end latency that is less than $d_{\max}$. Taking sleep latency to be the dominant component of end-to-end latency, the minimum average per-hop rate of progress required to satisfy the delay constraint is then given by:

$$S_{min} = \frac{P_{\max}}{d_{\max}}, \tag{3.6}$$

where $P_{\max}$ is the maximum multihop progress ('distance') from any node to the fusion center. Consequently, the selection process in node v_i is to find the set of forwarders F_i and the associated duty cycles α_{ij} $\forall v_j \in F_i$ such that: (i) $S_i \geq S_{min}$ to meet the rate of progress and delay constraints; (ii) maximum duty cycle required of each forwarder ($\max \alpha_{ij}$) is minimized, to prolong network connectivity and coverage.

3.4.1 Forwarding Set and Duty Cycle Selection

Recall that the forwarding set and associated duty cycles of each forwarder in A^2-MAC have to be selected such that: (i) rate of progress and delay constraints are met; and (ii) maximum duty cycle of each forwarder is minimized. By reducing the maximum duty cycle of any node, the likelihood that a node expends all of its energy much earlier than other nodes and subsequently causing network partitions is reduced.

We first present two lemmas that are useful in the selection process for the forwarding set and duty cycle of each node.

Lemma 1. *Let the set of candidate nodes $\aleph_i$ of node v_i be sorted in descending order of progress, from 1 to $|\aleph_i|$. The optimal set of forwarders $F_{opt(i)} \subseteq F_i$ that minimizes the maximum duty cycle requirement of v_i on its candidate nodes (i.e. $\min\max_{v_j \in \aleph_i} \alpha_{ij}$) is the first n_i forwarders with the largest progress, where n_i is a constant that is dependent on the progress of each forwarder in F_i.*

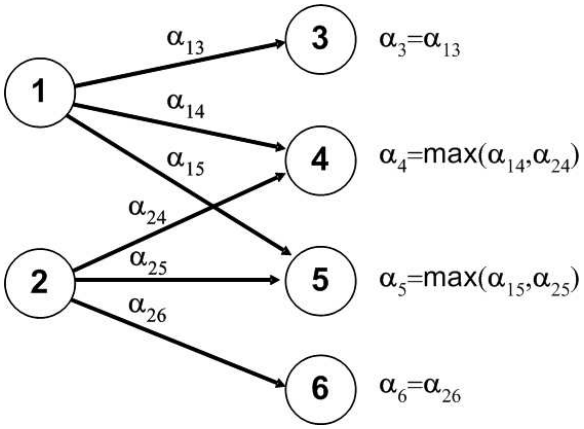


Figure 3.6: Computation of forwarding sets and duty cycles.

Proof. We proof Lemma 1 by contradiction. Suppose one of the forwarders in $F_{opt(i)}$ is not one of the n_i forwarders with the largest progress. We can swap this node with another node that is not in $F_{opt(i)}$, and which has larger progress. □

We consider the small network topology in Figure 3.6, where:

- node N1 has three candidate forwarding nodes such that $\aleph_1 = \{N3, N4, N5\}$ with corresponding progresses $p_{13} = 1$, $p_{14} = 0.9$ and $p_{15} = 0.2$; and
- node N2 has three candidate forwarding nodes such that $\aleph_2 = \{N4, N5, N6\}$ with corresponding progresses $p_{24} = 1$, $p_{25} = 0.75$ and $p_{26} = 0.5$.

Based on Lemma 1, the possible forwarding sets for N1 are $\{N3\}$, $\{N3, N4\}$ and $\{N3, N4, N5\}$ when $n_1 = 1$, $n_1 = 2$ and $n_1 = 3$ forwarders (with largest progress) respectively. Similarly for N2, the possible forwarding sets are $\{N4\}$, $\{N4, N5\}$ and $\{N4, N5, N6\}$ when $n_2 = 1$, $n_2 = 2$ and $n_2 = 3$ forwarders (with largest progress) respectively.

Lemma 2. *To meet the per-hop rate of progress constraint S_{min}, the maximum duty cycle requirement of v_i on its candidate nodes is minimized among all forwarders (i.e.* $\min\max_{v_j \in \aleph_i} \alpha_{ij}$*) iff their associated duty cycles requirements from v_i are the same, i.e. $\alpha_{ij} = \alpha_{ik}$ $\forall v_j, v_k \in F_i$.*

Proof. Again, we prove Lemma 2 by contradiction. Suppose that the duty cycles required of the forwarding nodes in F_i are not the same, i.e. $\exists \alpha_{ik} \neq \alpha_{ij}$ for some $v_j, v_k \in F_i$. Let v_j be the forwarder with the largest duty cycle requirement in F_i such that $\alpha_{ij} > \alpha_{ik}\ \forall v_k \in F_i$. If v_j has the largest progress among all the other forwarders such that $p_{ij} > p_{ik}$, then α_{ij} can always be reduced and the duty cycle requirement of another node v_k with smaller progress can be increased to ensure that $S_i \geq S_{\min}$. Conversely, if v_j is not the node with the largest progress such that $p_{ij} < p_{jk}$, then we can reduce α_{ij} by increasing the duty cycle requirement α_{ik} of another node v_k with larger progress. □

Based on Lemma 2, the minimum duty cycles required for each of the forwarding set combinations can be computed using Equations 3.3, 3.4 and 3.5. We assume that the minimum average per-hop rate of progress $S_{\min} = 2$ and (normalized) number of slots in each cycle $n_s = 1$ in Figure 3.6. We then consider the forwarding set {N3, N4} of Node N1 with $n_1 = 2$ forwarders. The average sleep latency is computed using $T_1 = \frac{1}{\alpha_{13}+\alpha_{14}+1}$ and the average per-hop progress is given by $P_1 = \frac{p_{13}\cdot\alpha_{13}+p_{14}\cdot\alpha_{14}}{\alpha_{13}+\alpha_{14}} = \frac{\alpha_{13}+0.9\alpha_{14}}{\alpha_{13}+\alpha_{14}}$, where $\alpha_{13} = \alpha_{14}$. Since $S_1 = \frac{P_1}{T_1} \geq S_{\min}$ is required to satisfy application constraints, $\frac{\alpha_{13}+0.9\cdot\alpha_{14}}{\alpha_{13}+\alpha_{14}} \cdot \frac{\alpha_{13}+\alpha_{14}+1}{1} \geq 2$. Solving this gives us $\alpha_{13} = \alpha_{14} = 0.5526$.

As illustrated in Table 3.1 and Figure 3.7, the minimum duty cycle requirements for the three forwarding set combinations {N3}, {N3, N4} and {N3, N4, N5} are $\alpha_{13} = 1$, $\alpha_{13} = \alpha_{14} = 0.5526$ and $\alpha_{13} = \alpha_{14} = \alpha_{15} = 0.619$ respectively. Hence, the minimum duty cycle is obtained when only {N3, N4} are used as forwarders. As N5 is not selected as a forwarding node of N1, the duty cycle requirement of N5 by N1 is subsequently set to be $\alpha_{15} = 0$.

As shown in Table 3.2, the minimum duty cycle requirements of node N2 are computed to be $\alpha_{24} = 1$, $\alpha_{24} = \alpha_{25} = 0.6429$ and $\alpha_{24} = \alpha_{25} = \alpha_{26} = 0.5556$ respectively for the forwarding sets {N4}, {N4, N5} and {N4, N5, N6}. In this case, all three forwarders should be used to obtain the minimum duty cycle. As

Table 3.1: Forwarding set and corresponding duty cycle requirements for N1.

Forwarding set	Duty cycle requirements
{N3}	$\alpha_{13} = 1$
{N3,N4}	$\alpha_{13} = \alpha_{14} = 0.5526$
{N3,N4,N5}	$\alpha_{13} = \alpha_{14} = \alpha_{15} = 0.619$

Table 3.2: Forwarding set and corresponding duty cycle requirements for N2.

Forwarding set	Duty cycle requirements
{N4}	$\alpha_{24} = 1$
{N4,N5}	$\alpha_{24} = \alpha_{25} = 0.6429$
{N4,N5,N6}	$\alpha_{24} = \alpha_{25} = \alpha_{26} = 0.5556$

the duty cycles required of forwarder N4 by nodes N1 and N2 are different, the final duty cycle of N4 is given by $\alpha_4 = \max\{\alpha_{14}, \alpha_{24}\} = 0.5556$[2].

Notice that when nodes are uniformly distributed in the network, n_i tends to approach $|\aleph_i|$ such that all the candidate nodes are included in the forwarding set F_i. However, when the progresses of the candidate nodes have high variations, nodes with smaller progresses are not included in the forwarding set, resulting in smaller values of n_i (as in the case of node N1).

By providing guidelines on how the forwarding set and duty cycle of each node should be selected to achieve the objectives of A^2-MAC, Lemmas 1 and 2 allow the search space of the combinations of forwarding sets and duty cycles of each node to be substantially reduced. Initially, all the nodes in the candidate set $\aleph_i$ are sorted in descending order of progress. Each candidate node is then incrementally added into the forwarding set F_i. We use $F_i^{(\varphi)}$ to denote the forwarding set containing the first φ nodes in $\aleph_i$ with the most progress, where $1 \leq \varphi \leq |\aleph_i|$. For each forwarding set $F_i^{(\varphi)} \subseteq \aleph_i$, the minimum α_{ij} required to ensure that $S_i \geq S_{min}$ is computed. The forwarding set with the smallest duty cycle requirement α_{ij} is considered to be optimal for node v_i. The algo-

[2]Note that N4 retains the duty cycle requirement α_{14} by forwarding less frequently for N1 at a rate of $\frac{\alpha_{14}}{\alpha_4}$ during its active slots.

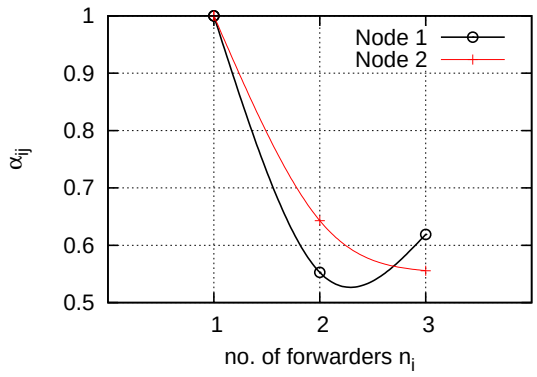


Figure 3.7: α_{ij} versus n_i for two different nodes N1 and N2.

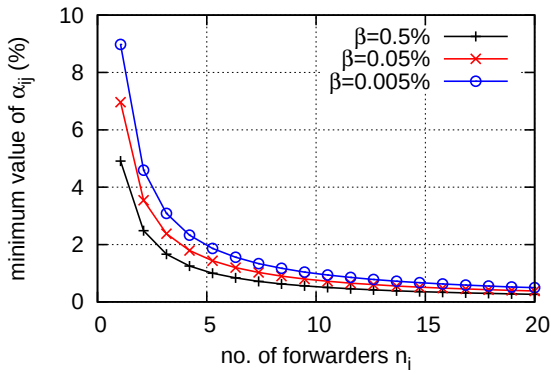


Figure 3.8: Minimum α_{ij} required to ensure that at least 1 forwarder is awake with $(1-\beta)\%$ within a cycle.

rithmic complexity for each node v_i to obtain its forwarding set and duty cycle requirements is in the order of $\Theta(\aleph_i)$.

3.4.2 Bounding the Maximum Sleep Latency

We note that when $\sum_{v_j \in F_i} \alpha_{ij}$ is small, the maximum sleep latency T_i for v_i can be arbitrarily large. T_i can be bounded by ensuring that the probability of having *no* forwarders active throughout a cycle with n_s time slots is less than a specified QoS threshold β $(0 < \beta < 1)$, such that:

$$\mathbb{P}_n = [\prod_{v_j \in F_i} (1 - \frac{\alpha_{ij}}{n_s})]^{n_s} \leq \beta. \tag{3.7}$$

Figure 3.8 illustrates the minimum average α_{ij} (as a percentage of n_s) required using varying values of n_i, using Equation 3.7. As the number of forwarders n_i increases, the duty cycle requirement of each node decreases exponentially.

3.4.3 The Adaptation Algorithm

The adaptation algorithm forms the core design of A^2-MAC, and allows each node v_i to compute: (i) its forwarding set F_i; (ii) duty cycles α_{ij} required **of** each forwarder $v_j \in F_i$; and (iii) its own duty cycle α_i based on the requirements **from** its upstream nodes. Adaptation is performed during network initialization as well as topological changes (e.g. node failure or mobility).

Initially, the duty cycles of all nodes are considered to be **undetermined**; for brevity, we refer to such nodes as 'undetermined' nodes. Similarly, a node that has computed its duty cycle in subsequent computations is referred to as a '**determined**' node. As such, an arbitrary node v_i can always divide its candidate set $\aleph_i$ into two disjoint sets such that $\aleph_i = \aleph_i^u \bigcup \aleph_i^d$, where $\aleph_i^u$ denotes the set of undetermined candidates and $\aleph_i^d$ denotes the set of determined candidates. It is trivial to see that during network initialization, $\aleph_i^u = \aleph_i$ and $\aleph_i^d = \emptyset$.

Each execution of the adaptation algorithm proceeds in bi-phase rounds. Algorithm 1 summarizes how, in the first phase of every round, a node v_i with $S_i < S_{\min}$ and undetermined candidate nodes computes: (i) its forwarding set F_i; and (ii) duty cycle requirements α_{ij} **of** each forwarder $v_j \in F_i$. These computations are based on the two lemmas presented in Section 3.4.1.

In each iteration of the `while` loop, the candidate node v_b that has the largest progress within the interim set of undetermined candidates Q_i^u is added to the current forwarding set F_i^φ (Lines 5 - 7). The interim set of undetermined candidates is then updated to exclude v_b (Line 8). Based on the current forwarding set F_i^φ, the minimum duty cycle requirement $\alpha_{ij}^{(\varphi)}$ is computed (Line 9) while ensuring that average per-hop rate of progress $S_i \geq S_{\min}$ (Equation 3.5) and $\mathbb{P}_n \leq \beta$ (Equation 3.7). The loop exits when the duty cycle requirements incorporating

Algorithm 1 Computation of F_i and α_{ij} by v_i in each round.

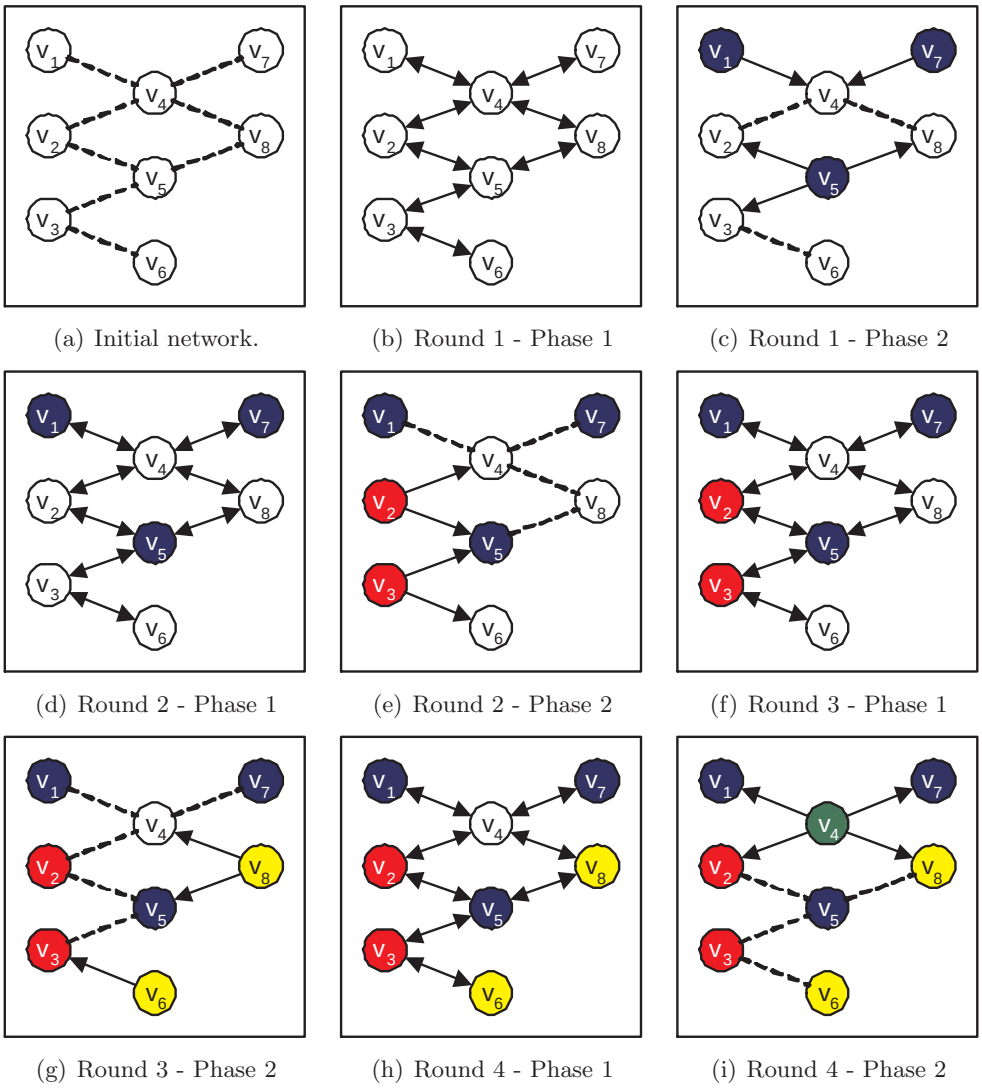
Require: $S_i < S_{\min}$; and $\aleph_i^u \neq \emptyset$.

1: **Input:** set of undetermined candidates $\aleph_i^u$, set of determined candidates $\aleph_i^d$, set of candidates $\aleph_i = \aleph_i^u \bigcup \aleph_i^d$, progress p_{ij} $\forall v_j \in \aleph_i$, duty cycle α_{ij} $\forall v_j \in \aleph_i^d$
2: **Variable:** $\varphi = |\aleph_i^d|$; current forwarding set $F_i^{(\varphi)} = \aleph_i^d$; interim set of undetermined candidates $Q_i^u = \aleph_i^u$; duty cycles of undetermined candidates $\alpha_{ij} = 0$ $\forall v_j \in \aleph_i^u$
3: **Output:** forwarding set F_i and duty cycle requirement α_{ij} $\forall v_j \in F_i$
4: **while** $Q_i^u \neq \emptyset$ **do**
5: $b = \underset{j}{\operatorname{argmax}}\ p_{ij},\ v_j \in Q_i^u$
6: $\varphi = \varphi + 1$
7: $F_i^{(\varphi)} = F_i^{(\varphi)} \bigcup \{v_b\}$
8: $Q_i^u = Q_i^u \setminus \{v_b\}$
9: Compute $\min \alpha_{ij}^{(\varphi)}$ using $F_i^{(\varphi)}$ such that $S_i \geq S_{\min}$ & $\mathbb{P}_n \leq \beta$
10: **end while**
11: $\phi = \underset{\varphi}{\operatorname{argmin}}\ \alpha_{ij}^{(\varphi)}$
12: $F_i = F_i^{(\phi)}$
13: $\alpha_{ij} = \alpha_{ij}^{(\phi)}$ $\forall v_j \in \aleph_i^u \bigcap F_i$
14: $\alpha_{ij} = 0$ $\forall v_j \in \aleph_i \setminus F_i$

each of the undetermined candidate nodes $\aleph_i^u$ have been computed (Line 4). The final forwarding set F_i and duty cycle requirement α_{ij} in the current round is the configuration that provides the minimum duty cycle requirements.

A key feature of A^2-MAC is that it exploits higher duty cycles of determined nodes to reduce the duty cycles requirements of (additional) undetermined candidate nodes. This is done by including all determined candidate nodes $\aleph_i^d$ into the current forwarding set $F_i^{(\varphi)}$ whenever Algorithm 1 is executed (Line 2). For instance, considering the network topology in Figure 3.6, once the larger α_4 value of 0.5556 is selected, α_3 can be reduced slightly from 0.5526 to 0.551.

Algorithm 2 summarizes the second phase of every round, whereby each undetermined node v_j computes its *interim* duty cycle $\widehat{\alpha}_j$ based on the duty cycle requirements **from** its upstream nodes (computed from the first phase - Algorithm 1). The undetermined node with the largest interim duty cycle among all its undetermined neighbors then fixes its duty cycle to be that of the computed interim, and is thereafter known as a 'determined' node. The next round then



(a) Initial network. (b) Round 1 - Phase 1 (c) Round 1 - Phase 2

(d) Round 2 - Phase 1 (e) Round 2 - Phase 2 (f) Round 3 - Phase 1

(g) Round 3 - Phase 2 (h) Round 4 - Phase 1 (i) Round 4 - Phase 2

Figure 3.9: Running behavior of the adaptation algorithms in a small network.

commences, until all the nodes in the network become determined. The adaptation algorithm is guaranteed to terminate, as at least one undetermined node becomes determined in each round. Assuming the worst case scenario whereby only one node becomes determined in each round, a total of n rounds are required to terminate the algorithm, where n is the total number of nodes in the network. Hence, the overall complexity of the adaptation algorithm is in the order of $O(n \cdot \max |\aleph_i|)$.

Figure 3.9 illustrates the running behavior of the adaptation algorithms (Al-

Algorithm 2 Determination of duty cycle α_j of each undetermined node v_j.

1: **Input:** set of neighbors Z_j; duty cycle requirements α_{ij} from upstream nodes $v_i \in Z_j \setminus \aleph_j$
2: **Variable:** interim duty cycle $\widehat{\alpha}_j$; determined = `FALSE`
3: **Output:** determined; duty cycle α_j (if determined = `TRUE`)
4: $\widehat{\alpha}_j = \max_{v_i \in Z_j \setminus \aleph_j} \alpha_{ij}$
5: Broadcast value of $\widehat{\alpha}_j$ to set of neighbors Z_j
6: Receive values of $\widehat{\alpha}_i$ from $v_i \in Z_j$
7: **if** $\widehat{\alpha}_j > \max_{v_i \in Z_j} \widehat{\alpha}_i$ **then**
8: $\alpha_j = \widehat{\alpha}_j$
9: determined = `TRUE`
10: **end if**

gorithms 1 and 2) in a small network with 8 nodes, v_1 to v_8. As illustrated in Figure 3.9(a), all the nodes are undetermined during network initialization. Recall that the adaptation algorithms proceeds in bi-phase rounds. After Algorithm 1 is executed as part of Round 1 Phase 1, each node in the network broadcasts its forwarding set and duty cycle requirements in Figure 3.9(b). After each undetermined node has computed and broadcasted its interim duty cycle in Round 1 Phase 2, nodes v_1, v_5 and v_7 become determined as they have the largest interim duty cycles within their respective undetermined neighbors. These newly determined nodes then broadcast their (fixed) duty cycles to their neighbors in Figure 3.9(c). The process repeats until all the nodes in the network become determined. Note that in each round, at least one node becomes determined; however, each undetermined node that becomes determined in the same round must be at least two hops away from one another.

3.5 Performance Evaluation

We evaluate the performance of A^2-MAC using GloMoSim [121], a simulator for large-scale wireless networks. The results shown in this section are averaged over 20 runs. The size of each data packet is 60 Bytes, and all traffic arrivals follow a Poisson distribution. The transmission range of each node is approximately 60

Table 3.3: Simulation Parameters

Parameter	Value
Transmission range	$\approx$ 60 meters
Bandwidth	250 kbps
SLEEP mode	0.001 mA
IDLE mode	0.426 mA
RX mode	19.7 mA
TX mode	11.0 mA
Control packet duration	$\approx$ 0.5 ms
A^2-MAC time slot length τ	20 ms
A^2-MAC cycle length	2 s
Network size	100 to 300 nodes
Delay constraint $d_{\max}$	{2, 3, 4, 5, 6} s

meters, and the terrain has a size of 250 m $\times$ 250 m. The fusion center is placed at the top right-hand corner of the terrain. The Maximum Forward Progress (MFP) routing protocol [120] is used to forward data to the fusion center via multihops. Some of the common simulation parameters are listed in Table 3.3[3].

We evaluate the performance of A^2-MAC with: (i) X-MAC [58], a well-known energy efficient asynchronous unicast MAC protocol; and (ii) opt-MAC [115], which is optimal among approaches using the *same* duty cycle for all the nodes and provides an average delay constraint. Sections 3.5.1 to 3.5.3 assume that energy expended in packet transmission is negligible as compared to energy expended through long periods of idle listening. In Sections 3.5.4 and 3.5.5, we consider higher traffic loads where transmissions incur significant energy.

3.5.1 Delay Tradeoffs

We vary the delay constraint $d_{\max}$ from 2s to 6s and study the delay tradeoffs of the three MAC protocols (A^2-MAC, X-MAX and opt-MAC) in Figure 3.10. 150 nodes are uniform-randomly distributed in the network, yielding an average

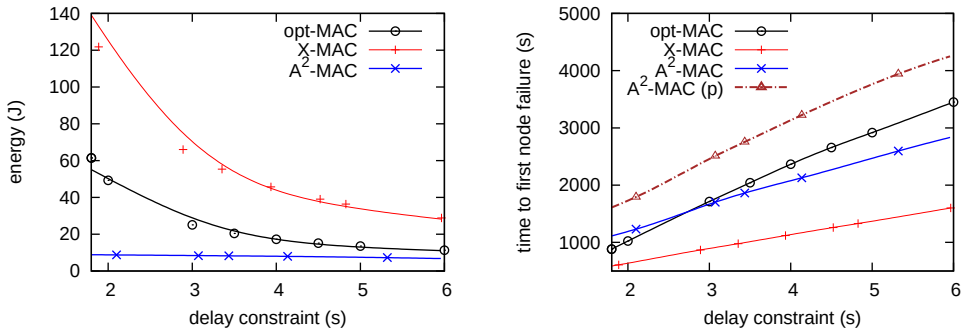
[3]Based on Chipcon CC2420 RF transceiver specifications.

node degree of approximately 20. As $d_{\max}$ increases, nodes are able to sleep longer while satisfying the delay constraint, leading to lower duty cycles and hence lower per-node energy consumption, as shown in Figure 3.10(a). A^2-MAC achieves better energy-delay tradeoffs particularly for smaller values of $d_{\max}$, which reflects tighter delay constraints and higher energy expenditure. With the use of adaptive duty cycles, A^2-MAC allows each node to vary its duty cycle based on local network topology, thereby reducing energy consumption wherever possible. In contrast, X-MAC and opt-MAC assign the same (maximum) duty cycle to all the nodes, resulting in higher overall energy consumption. The complementary use of anycast forwarder selection allows A^2-MAC to reduce its end-to-end delay significantly by forwarding data quickly to any neighbor that is awake in the forwarding set, instead of waiting for a particular forwarder to be awake. Due the the large number of neighbors (≈ 20) in the network, each node is already using a very small duty cycle, resulting in a very small decrease in the energy consumption incurred by A^2-MAC as $d_{\max}$ increases.

As A^2-MAC does not *globally* optimize the time to the first node failure, it performs slightly worse than opt-MAC (which is optimized for this aspect) for higher $d_{\max}$ values in Figure 3.10(b). In opt-MAC and X-MAC, nodes are assigned the same duty cycles and fail at the same rate; in A^2-MAC, nodes fail gracefully over time. Consequently, the time to network partition for A^2-MAC - denoted as A^2-MAC(p) - exceeds the time to first node failure of opt-MAC by 20% to 50%, as the network remains connected even when some nodes in the (typically dense) sensor network has failed.

3.5.2 Connectivity and Coverage

Figure 3.11 illustrates the network connectivity and coverage over time with $d_{\max}$ = {2, 5}s, in a network of 150 nodes. The percentage connectivity is the ratio of nodes that remain alive and connected to the fusion center relative to total number of nodes in the network. The percentage coverage is the ratio of the



(a) Energy (b) Time to First Node Failure

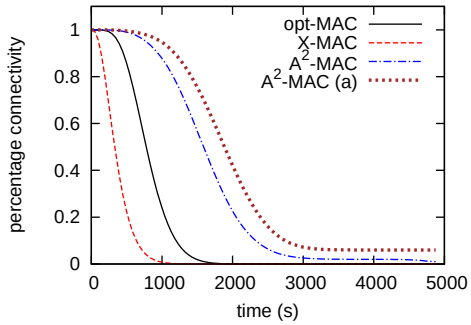
Figure 3.10: Delay tradeoff under varying delay constraints.

terrain within the range of any connected and alive node relative to the initial coverage area, under the assumption that the sensing and communication ranges of a node are equal. These two metrics provide measures of how quickly network connectivity and coverage deteriorate over time as nodes fail due to energy drain.

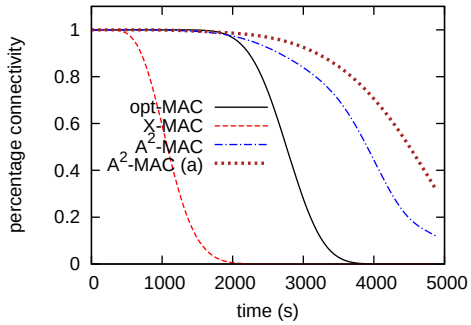
Due to the unicast nature of X-MAC, nodes do not exploit the redundancy of neighbors to reduce duty cycles; hence its percentage connectivity deteriorates very quickly over time. Although opt-MAC utilizes an anycast mechanism to minimize the maximum duty cycle required of its node to meet the delay constraint, its network connectivity still deteriorates quickly as all nodes use the same maximum required duty cycle. Figures 3.11(a) and 3.11(b) show that A^2-MAC has the best percentage connectivity as it: (i) minimizes the local maximum duty cycle; and (ii) adaptively assigns (different) duty cycles to each node based on its local topology.

We note that the percentage of *alive* nodes in A^2-MAC - denoted as A^2-MAC(a) - is higher than the percentage connectivity; this indicates that there are nodes that are alive but have lost connectivity to the fusion center. Although these nodes are disconnected from the fusion center, they are potentially useful as they can transmit data to the fusion center when the network is repaired, or through techniques such as message ferrying.

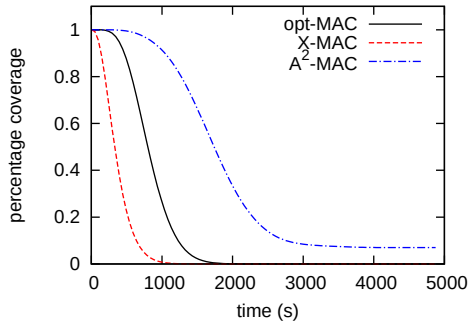
The higher network connectivity in A^2-MAC allows it to achieve better per-



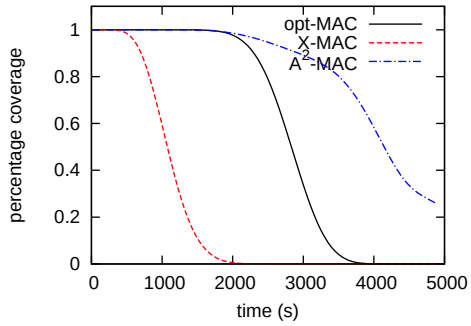
(a) Percentage connectivity over time with $d_{\max} = 2$s



(b) Percentage connectivity over time with $d_{\max} = 5$s



(c) Percentage coverage over time with $d_{\max} = 2$s



(d) Percentage coverage over time with $d_{\max} = 5$s

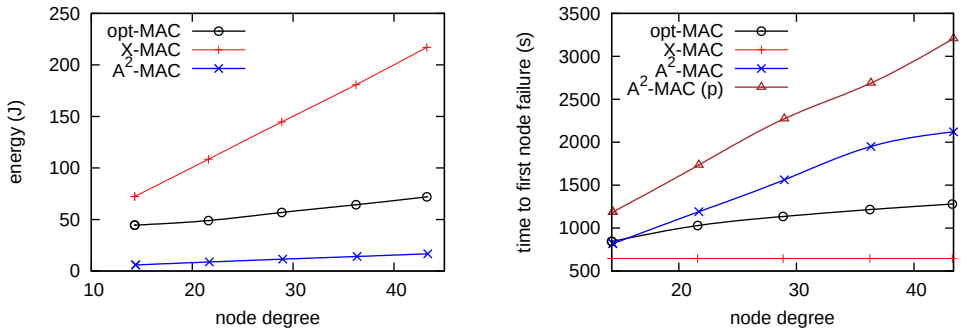
Figure 3.11: Performance of opt-MAC, X-MAC and A^2-MAC under varying delay constraints.

centage coverage over time than opt-MAC and X-MAC in Figures 3.11(c) and 3.11(d). Notice that in A^2-MAC, a small percentage of the nodes remain connected and cover a small proportion of the network for a long time. These nodes are close to the fusion center and have few upstream nodes. Their energy consumption is extremely low as they perform minimal data forwarding for their neighbors.

3.5.3 Random Topology with Varying Network Densities

The performance of the MAC protocols in networks with varying densities and delay constraint $d_{\max} = 2$s is shown in Figure 3.12. The network size is varied from 100 to 300 nodes such that the node degree varies between 15 to 45.

Figure 3.12(a) indicates that energy consumption per node increases with



(a) Energy consumption per node.

(b) Time to first node failure.

Figure 3.12: Performance with varying network densities and $d_{\max} = 2$.

increasing node degree. As X-MAC employs a unicast mechanism, it does not exploit the availability of increased redundancy (or neighbors) to reduce the duty cycles of nodes, resulting in high energy consumption. By utilizing larger forwarding sets as node degree increases, both opt-MAC and A^2-MAC can achieve low energy consumption through the use of anycast forwarding mechanisms. The latter consumes the least energy as it tends to select neighbors that provide more progress as forwarders, and allows nodes to adapt (lower) their duty cycles according to local topologies.

The time to the first node failure of X-MAC is independent of node degree, as illustrated in Figure 3.12(b), as each node uses the same duty cycle and has a single fixed neighbor in its forwarding set. Through the use of the anycast forwarding mechanism, A^2-MAC and opt-MAC can utilize more forwarders with increasing node degrees, resulting in lowered duty cycles and longer times to first node failure. A^2-MAC can achieve longer time to first node failure than opt-MAC as it does not require all the nodes to use the same (maximum) duty cycle. In addition, it is able to achieve longer time to network partitions, as its anycast and adaptive mechanisms maximize the benefit of network redundancy.

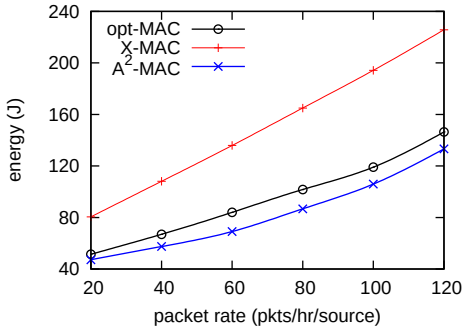


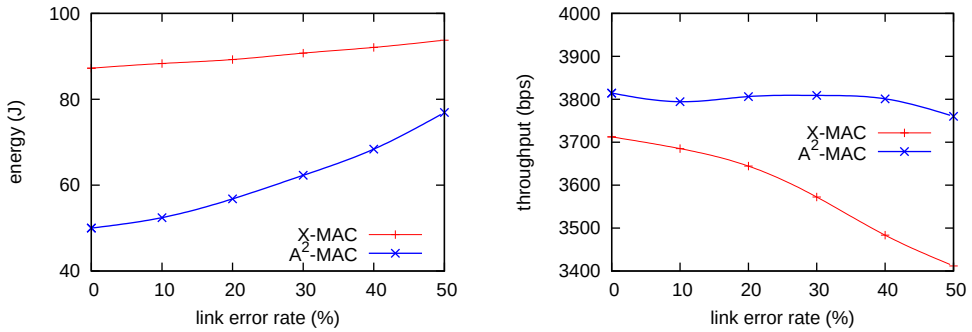
Figure 3.13: Performance of opt-MAC, X-MAC and A^2-MAC with varying traffic loads.

3.5.4 Random Topology with Varying Traffic Loads

We evaluate the performance of the MAC protocols with varying traffic loads and a delay constraint of $d_{\max} = 300$ms. 50% of the nodes are randomly selected to generate data packets; the traffic arrival rate for *each* of these selected source node ranges from 20 to 120 pkts/hour. There are 100 uniform-randomly distributed nodes in the network, yielding an average node degree of 15.

As traffic load increases, the energy consumed by all the MAC protocols increases correspondingly in Figure 3.13, due to increased data transmissions. As X-MAC employs a unicast technique, each transmitting node has to continuously transmit strobed preambles until the pre-selected next-hop is awake, even though there may exist multiple neighboring nodes with positive progress towards the fusion center. This results in excessive overheads and subsequently, higher energy consumption per node.

With the use of anycast during the data forwarding process, both A^2-MAC and opt-MAC are able to reduce the number of control packets transmitted per data packet, and subsequently reducing the overall energy consumption. Overall, the average energy savings of A^2-MAC (and opt-MAC) over X-MAC is approximately 40% to 50%.



(a) Energy Consumption (b) Throughput

Figure 3.14: Performance with intermittent link connectivity.

3.5.5 Random Topology with Intermittent Link Connectivity

Figure 3.14 studies the robustness of the MAC protocols in a network with intermittent link connectivity. 50% of the 100 nodes in the network are randomly selected as data sources and 10% of the links are randomly selected to have link error rates ranging from 0% to 50%. A link error rate of 0% indicates the absence of artificially induced link errors, and the delay constraint is set to $d_{\max}$ = 300ms when there is no link error. As opt-MAC utilizes a similar anycast approach as A^2-MAC, its performance in intermittently connected networks is similar to A^2-MAC and is not shown.

Generally, the number of retransmissions required for each data packet increases with increasing link error rates. This results in the corresponding increase in energy consumption in Figure 3.14(a). In the presence of packet losses caused by link errors, X-MAC retransmits unsuccessfully over the same link with poor quality, resulting in high energy consumption and low throughput. In contrast, when transmission to a particular neighbor is unsuccessful, A^2-MAC leverages on the availability of multiple forwarding nodes to counter link failures by transmitting data to another forwarder in subsequent time slots. By dynamically selecting the next-hops forwarders based on prevailing network conditions, A^2-MAC is more resilient to intermittent link failures, enabling it to achieve higher and more consistent throughputs than X-MAC, as illustrated in Figure 3.14(b).

3.5.6 Discussion

In this section, we have evaluated A^2-MAC by varying various network parameters such as delay constraints, network sizes, traffic loads and link error rates. The performance of A^2-MAC has been compared with: (i) X-MAC, an asynchronous MAC protocol which is unicast in nature; and (ii) opt-MAC, an anycast MAC protocol that is optimal among approaches that utilize the same duty cycle for all the nodes in the network.

Our simulation studies have verified the significance of utilizing an anycast and adaptive mechanism for forwarder selection, especially when duty cycles are low and in the presence of intermittent links (which are inherent in practical network scenarios). Subsequently, both A^2-MAC and opt-MAC are able to achieve superior performance over X-MAC. In addition, while opt-MAC can achieve slightly longer times to first node failure with utilization of end-to-end connectivity information, significantly larger proportion of nodes remain connected in A^2-MAC over time, even though only local information is used. In sensor networks where there is often sufficient node redundancy resulting from dense node deployments, we believe that the number of connected nodes over time is a more relevant measure of network lifetime than time to first node failure.

3.6 Summary

The severe energy limitations in sensor motes accentuate the need for energy efficient MAC protocols. However, duty cycling incurs higher latencies as transmitters have to wait for forwarders to be awake before communication can commence. In this chapter, we detail the design of A^2-MAC, an adaptive, anycast-based MAC protocol that utilizes an asynchronous random wakeup schedule, anycast mechanism as well as adaptive forwarding set selection and duty cycle selection. A^2-MAC adapts its duty cycle and forwarding set based on local network topology and a given delay constraint to achieve energy efficiency with low

latencies. It can also achieve better connectivity and coverage, and significantly outperforms existing asynchronous sensor MAC protocols.

As A^2-MAC is designed for generic energy constrained networks, it can be used for both periodic monitoring and event driven sensor applications. There exists additional potential developments in the adaptation for A^2-MAC, for example, adapting the duty cycles depending on their distance from the fusion center and traffic loads.

In the next two chapters, we look into the design of energy efficient communication protocols for event driven systems (as these generally have more stringent application requirements than periodic monitoring systems), and which also incorporate the key properties of A^2-MAC into their operations.

Chapter 4

Information Quality Aware Routing

This chapter describes IQAR [34], an Information Quality Aware Routing protocol designed for the class of event driven sensor networks. It utilizes data aggregation and IQ-awareness to achieve energy efficiency while meeting IQ constraints.

4.1 The Case for Data Aggregation and/or Fusion

Event driven sensor networks are deployed specifically for the detection of phenomena of interest (PoI). Such networks have convergecast traffic characteristics [122] and sensory data is typically generated only when a PoI is detected. Upon the occurrence of a PoI (such as a fire hazard [10] or an elderly person falling down in a monitored home environment [11]), multiple sensors may be activated concurrently. As sensor networks tend to have dense deployments, this can lead to severe data implosion and redundancy [35], and subsequently excessive energy expenditure.

To reduce traffic load, as well as mitigate the effects of congestion and medium access contention, data aggregation and/or fusion techniques are often

used to combine data from multiple sensor sources enroute [123] [124] [91] [125]. These in-network processing techniques exploit the presence of spatio-temporal correlation [126] among sensory data, based on the following principles:

Spatial Correlation: Nodes that are of the same geographical proximity tend to sense the same physical phenomenon; hence, it is not necessary for all of these nodes to send data back to the fusion center all the time.

Temporal Correlation: The physical environment sensed by each sensor node is unlikely to have drastic changes within small time intervals; thus, it is unnecessary for the sensor to transmit all its data back to the fusion center at every sensing interval.

Generally, such aggregation/fusion techniques can be classified into two main categories, viz. structure-less and structured.

In structure-less techniques [91] [74], data aggregation occurs *opportunistically* only when data flows happen to meet at the same time at the same intermediate forwarding node. There is no deliberate attempt to delay any transmission or re-route packets such that these encounters take place. Consequently, structure-less approaches incur shorter delays when the network is lightly loaded, as data is forwarded towards the fusion center using an underlying shortest-path or least-cost routing algorithm. However, such approaches do not scale well with large networks or high traffic volumes as aggregation opportunities are not maximized.

In structured techniques [127] [128] [129] [130], routing paths are computed and maintained to allow efficient data aggregation. The routing path is influenced primarily by the amount of data reduction that can be achieved by data compression before it is forwarded to the fusion center. Such techniques incur relatively higher overheads to maintain the network structure, and are associated with a delay factor, as intermediate forwarding nodes have to wait for upstream nodes to send data to them, before aggregating these data packets and forward-

ing them to the fusion center. Nevertheless, structured techniques can achieve significant energy savings, as data is maximally aggregated along the forwarding paths. They are well-suited for sensor networks with slow-varying traffic characteristics, such as periodic monitoring. Many structured schemes adopt a clustering approach [87] [88] [131], whereby sensory data is first transmitted to a cluster head for aggregation before being forwarded to the fusion center.

Although these existing schemes are able to reduce transmission costs and energy consumption, they do not take the information quality of data into account during the aggregation/fusion process. This can lead to one of the following situations:

- Over-provisioning of IQ and high data redundancy at fusion center, resulting in unnecessary energy expenditure.
- Under-provisioning of IQ at fusion center, resulting in loss in event *detection accuracy*.

In the next section, we outline how the integration of IQ-awareness in data aggregation/fusion can achieve energy efficiency in event driven sensor networks, without compromising the information quality of data at the fusion center.

4.2 The Case for Information Quality Awareness

In event driven sensor networks, there exists an obvious tradeoff between energy efficiency and information quality with the use of data aggregation and/or fusion schemes [132]. Through the collection of more data from sensor sources, higher information quality can be achieved at the fusion center, at the cost of higher energy expenditure. Conversely, although energy consumption is reduced when less data is collected from sensor sources, the information quality is lower at the fusion center. This is equivalent to a loss in event *detection accuracy*, which may result in overall lower system and network reliability.

4.2.1 Existing IQ-Aware Schemes

Unlike aggregation based routing schemes, IQ-aware routing schemes consider the information content of data during data aggregation and forwarding. Information directed approaches such as IDSQ and CADR [133] [134] [135] utilize energy efficient techniques to handle data querying and routing, while minimizing delay and bandwidth consumption. At each step along the routing path, the neighboring node with the highest predicted information gain is selected to be the next-hop forwarder, and the fused data is transmitted along a single path to the fusion center as soon as it satisfies a given IQ threshold. However, these protocols are query-based and targeted at tracking applications.

Although IQ-aware schemes for event detection sensor networks have also been proposed in the literature [136] [137] [138] [139], their emphasis is on designing energy efficient hypothesis testing models to detect the presence of the PoI. In addition, most of these schemes are based on a centralized one-hop sensor network topology, and do not consider multihop routing to the fusion center. Consequently, there is a need for an IQ-aware multihop routing protocol for event detection in wireless sensor networks.

4.2.2 A NP-Hard Routing Problem

We address the problem of finding a least cost (minimum energy) routing tree that satisfies a given IQ constraint, to achieve energy efficiency in event detection sensor networks. It is noted that the optimal least cost routing solution is a variation of the classical **NP-hard** Steiner tree in graphs [72]. In the context of event driven sensor networks, the original Steiner tree problem is to find a Steiner Minimum Tree that spans the fusion center and entire set of activated sensor nodes (denoted as V_a) that detect the PoI. However, it is unnecessary and expensive for all the sensors in V_a to transmit their data to the fusion center.

Ideally, an IQ-aware event driven routing scheme only needs to aggregate suf-

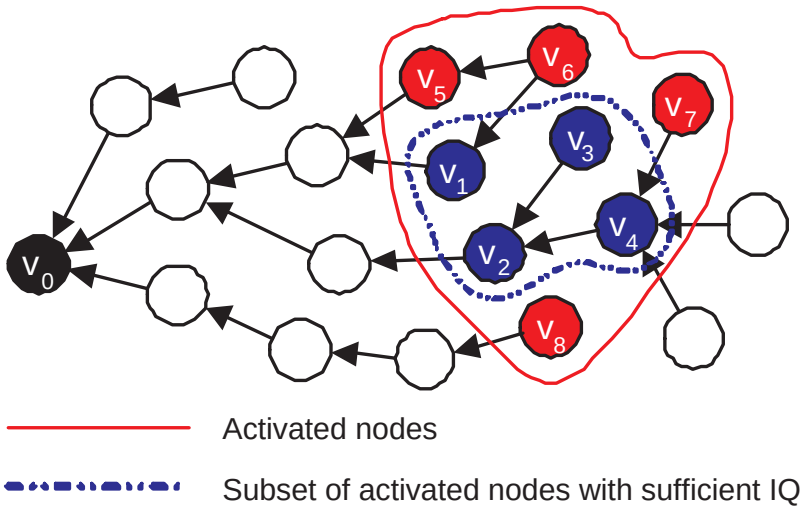


Figure 4.1: Event driven sensor network with set of activated nodes $V_a = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$. $V_\tau = \{v_1, v_2, v_3, v_4\}$ represents *one* possible subset of activated nodes that can detect PoI with sufficient IQ.

ficient sensory data from a subset of activated nodes $V_\tau \subseteq V_a$ to meet the IQ constraint. Considering the event driven sensor network in Figure 4.1, suppose sensor nodes v_1 to v_8 are activated when the PoI occurs, i.e. $V_a = \{v_1, v_2, ..., v_7, v_8\}$. Instead of aggregating data from all these nodes at the fusion center, nodes v_1, v_2, v_3 and v_4 represent *one* possible subset of activated nodes that can detect the PoI with sufficient IQ at the fusion center, such that $V_\tau = \{v_1, v_2, v_3, v_4\}$. This minimizes the cost (or energy) required to detect the PoI reliably in an event driven sensor network.

We refer to this as a ***subset-τ* Steiner Tree (SST) problem**[1]. Clearly, the latter is a harder problem than the Steiner tree, as the: (i) set of activated nodes V_a and IQ contribution of each activated node are not known *a priori* until the PoI occurs; and (ii) subset of activated nodes V_τ whose aggregated IQ meets the given constraint is not unique. Knowledge of the entire network topology and individual IQ contributions of each activated sensor node are essential to find the least-cost SST that satisfies the required IQ. Unfortunately, this incurs extensive computational, storage as well as communication overheads, and is not

[1]We define the SST problem formally in Section 4.3.6

a feasible approach in resource-constrained sensor networks.

We tackle the fundamental issues in constructing an optimal least-cost SST in an IQ-aware event driven sensor network by proposing: (i) a topology-aware histogram-based aggregation structure that encapsulates the cost of including the IQ contribution of each activated node, in a compact and efficient way; and (ii) a greedy heuristic to approximate and prune a least cost aggregation routing path. The proposed **IQ-Aware Routing** (IQAR) protocol constructs an initial distance-based aggregation tree that spans *all* the sensors in the sensor network. When a PoI occurs, activated sensors forward their data to the fusion center using the underlying pre-built distance-based aggregation tree. At each hop along the routing path, the IQ contributions of each activated sensor and its downstream (forwarding) nodes are discretized and incorporated together, to form a topology-aware histogram-based aggregation structure.

When data packets (with incorporated histograms) reach the fusion center, it utilizes a greedy heuristic to prune the original aggregation tree such that: (i) aggregated IQ of the resulting pruned tree satisfies a given IQ constraint; and (ii) total cost of collecting data from activated nodes in the pruned tree is minimized. The pruning process is recursively executed at each forwarding node along the initial aggregation tree.

Activated sensor nodes that are not part of the pruned routing tree suppress their data for a time epoch. When the time epoch expires, activated nodes whose data have been suppressed resume the forwarding of their data to the fusion center. This allows the routing protocol to be adaptive to dynamic changes in the network and phenomena of interest. Our studies show that the performance of IQAR is upper bounded by a distance-based aggregation tree that collects data from all the activated nodes, and comparable to another IQ-aware routing protocol that uses an exhaustive brute-force search to approximate and prune the least-cost aggregation tree.

4.3 System Model

In an event-detection sensor network, the *information quality* of concern is related to the detection accuracy of the system. In this section, we detail how IQ is mapped onto the targeted detection and false alarm probabilities P_d and P_f using sequential detection [140] [141]. We also describe the Likelihood Ratio Test (LRT) [142], which has been shown to be the optimal detection scheme that maximizes detection probability.

The network is modeled as a graph $G = \{V, E\}$, where $V = \{v_0, v_1, v_2, ..., v_n\}$ denotes the set of n sensor nodes and fusion center v_0, and E denotes the set of edges (or links) between any two nodes. An edge $e_{ij} \in E$ represents the existence of a communication link between two arbitrary sensors v_i and v_j.

We let hypothesis H_1 denote the presence of a PoI in the sensor network; H_0 denotes the corresponding absence of the PoI. The probabilities $P(H_1) = p$ and $P(H_0) = 1 - p$, where $0 < p < 1$, are known *a priori*. Each node *independently* senses and collects data about the environment periodically. When conditioned upon the hypothesis H_i, $i \in \{0, 1\}$, sensor observations are assumed to be independently and identically distributed (i.i.d.) at each sensor as well as across sensors [141].

4.3.1 Event Detection at Sensor

The independent signal y_i observed by a node v_i is given by:

$$y_i = \begin{cases} w_i & \text{if } H_0 \text{ (PoI is absent);} \\ f(r_i) + w_i & \text{if } H_1 \text{ (PoI is present),} \end{cases} \tag{4.1}$$

where $w_i \sim \mathcal{N}(0, \sigma_w^2)$ is the white Gaussian noise seen by v_i that follows a normal distribution with mean 0 and standard deviation σ_w; r_i is the the distance between v_i and the PoI; and f is a function that monotonically decreases with increasing r_i. An example of such a function is the exponential sensing model

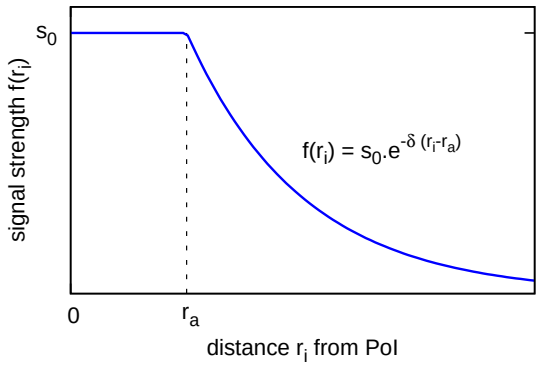


Figure 4.2: Signal strength $f(r_i)$ with $\alpha = 0.5$.

[143] [144] in Figure 4.2, which is defined by:

$$f(r_i) = \begin{cases} s_0 & \text{if } r_i \leq r_a; \\ s_0 \cdot e^{-\delta(r_i - r_a)} & \text{otherwise,} \end{cases} \tag{4.2}$$

where r_a is the sensing range below which the signal generated by the PoI starts to undergo attenuation; s_0 is the signal strength of the PoI when measured within a distance of r_a from the PoI; and δ is the sensing capacity decay (reflecting the attenuation of the signal generated by the PoI).

For each sampled signal y_i, v_i makes a **per-sample binary decision** $b_i \in \{0, 1\}$ such that:

$$b_i = \begin{cases} 0 & \text{if } y_i < \mathbb{T}_i; \\ 1 & \text{otherwise,} \end{cases} \tag{4.3}$$

where $\mathbb{T}_i$ is the per-sample threshold of v_i.

The per-sample probability of false alarm p_0^i by v_i is independent of its location, and given by [136]:

$$p_0^i = P(b_i = 1 | H_0) = Q(\frac{\mathbb{T}_i}{\sigma_w}), \tag{4.4}$$

where $Q(x)$ is the Gaussian Q-function of a standard normal distribution. The

corresponding per-sample probability of detection p_1^i (where $p_1^i > p_0^i$) at v_i is dependent on the distance r_i between v_i and the PoI, and given by:

$$p_1^i = P(b_i = 1|H_1) = Q(\frac{\mathbb{T}_i - f(r_i)}{\sigma_w}). \tag{4.5}$$

Although each node senses the environment at periodic intervals, it is infeasible for each of these samples to be transmitted to the fusion center v_0, due to limited network bandwidth and energy constraints of the sensors. Consequently, a data packet generated by v_i is transmitted to v_0 only if v_i detects the presence of a PoI and becomes **activated** (when $b_i = 1$).

4.3.2 Event Detection at Fusion Center

The role of the fusion center v_0 is to detect the presence of the PoI by making a **global binary decision** $\hat{H} = \{H_0, H_1\}$, based on the data that it has received from the set of activated nodes V_a. Let $B = \{b_1, b_2, ..., b_{|V_a|}\}$ be the set of per-sample binary decisions that v_0 receives from each activated node $v_a \in V_a$ in a time epoch. The optimal fusion rule for v_0 using data from *all* the activated nodes is the Likelihood Ratio Test (LRT) [141] [142]:

$$\Lambda(B) = \frac{P(b_1, b_2, ..., b_{|V_a|}|H_1)}{P(b_1, b_2, ..., b_{|V_a|}|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{1-p}{p}. \tag{4.6}$$

Recall that $p = P(H_1)$ is the *a priori* probability that the PoI is present.

The LRT can be interpreted in this way: If $\Lambda(B) \geq \frac{1-p}{p}$, then it is more likely that H_1 is true; otherwise, it is more likely that H_0 is true. In practice, $\frac{1-p}{p}$ is selected such that $P(\Lambda(B) < \frac{1-p}{p}) = \alpha$, where α is the probability that the PoI is not detected when it occurs. Hence, v_0 makes the decision that the PoI is present ($\hat{H} = H_1$) if $\Lambda(B) \geq \frac{1-p}{p}$, and the decision that the PoI is absent ($\hat{H} = H_0$) otherwise. Notice that for small values of *a priori* probability p, the likelihood ratio $\Lambda(B)$ required for the PoI to be detected is much larger than for

bigger values of p. As a numerical example, consider $p = 0.01$; then, the PoI can be detected only when $\Lambda(B) \geq \frac{1-0.01}{0.01} = 99$. In contrast, if $p = 0.1$, the PoI can be detected when $\Lambda(B) \geq \frac{1-0.1}{0.1} = 9$.

4.3.3 Sequential Detection

The event detection model in Section 4.3.2 requires data from all the activated nodes to be collected at the fusion center. This can incur excessive overheads and energy consumption, especially in dense networks where the number of activated nodes can be quite large.

To reduce the amount of data that is collected for v_0 to make an accurate global binary decision $\hat{H}$, we adopt the sequential detection model which is based on the Sequential Probability Ratio Test (SPRT) proposed by A. Wald [140]. In SPRT, the amount of data required is a random variable dependent on the prior data that has been obtained thus far.

Let $X_a = \{x_1, x_2, ..., x_{|V_a|}\}$ be the *sequence* whereby data is collected from each activated node $v_a \in V_a$. Using sequential detection, data acquisition can terminate at the earliest subsequence of fused local data $X_\tau = \{x_1, x_2, ..., x_\tau\} \subseteq X_a$ when the decision $\hat{H} = \{H_0, H_1\}$ can be made, thus minimizing the cost of data acquisition and PoI detection.

We denote the hopcount of an arbitrary node v_i as h_i. The neighboring node v_j with hopcount $h_j = h_i + 1$ and which uses v_i to forward packets to the fusion center is considered an **upstream** node of v_i; the set of upstream nodes of v_i is denoted as V_i^u. In the same manner, v_i is known as the **downstream** node of v_j. In Figure 4.3, v_1, v_2 and v_3 are the upstream nodes of v_0 such that $V_0^u = \{v_1, v_2, v_3\}$.

Since observations across sensor nodes are i.i.d., the **cumulative log-likelihood**

ratio S_0 at the fusion center v_0 is given by:

$$S_0 = \log \Lambda(B) = \log \prod_{i=1}^{|V_a|} \Lambda(b_i). \tag{4.7}$$

The corresponding cumulative log-likelihood ratio S_i at v_i comprises of its log-likelihood ratio and the cumulative log-likelihood ratios of each upstream node $v_j \in V_i^u$, such that:

$$S_i = \log \Lambda(b_i) + \sum_{v_j \in V_i^u} S_j. \tag{4.8}$$

The **stopping rule** $\gamma_i = \{0, 1\}$ is computed after each incorporated data from v_i, and is dependent on the targeted detection and false alarm probabilities P_d and P_f. It determines if the current sequence of incorporated data along the routing path is sufficient for the global decision $\hat{H}$ to be made at v_0, and is given by Wald's Equation [140]:

$$\gamma_i = \begin{cases} 0 & \text{if } \mathbb{A} < S_i < \mathbb{B}; \\ 1 & \text{otherwise,} \end{cases} \tag{4.9}$$

where $\mathbb{A} = \log(\frac{1-P_d}{1-P_f})$; and $\mathbb{B} = \log(\frac{P_d}{P_f})$. This stopping rule is considered to be optimal in sequential detection, as it results in the least possible amount of data required for decision making.

If $\gamma_i = 0$, the current sequence of data collected is insufficient for a global decision $\hat{H}$ to be made and more data samples have to be acquired. However, when $\gamma_i = 1$, the decision $\hat{H}$ can be made based on the current sequence of incorporated data, according to:

$$\hat{H} = \begin{cases} H_0 & \text{if } S_i \leq \mathbb{A}; \\ H_1 & \text{otherwise } (S_i \geq \mathbb{B}). \end{cases} \tag{4.10}$$

Hence, additional data need not be collected from other sensor nodes and data

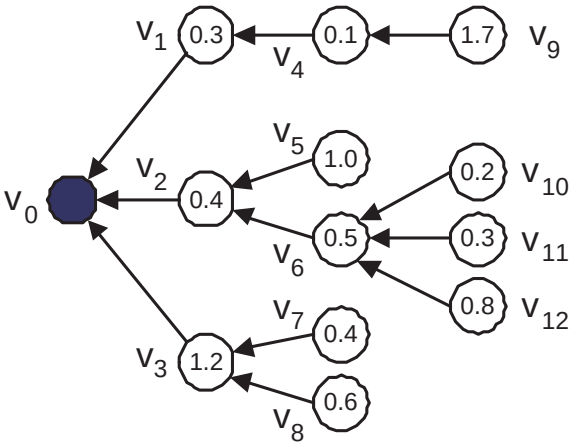


Figure 4.3: Fusion center v_0 with three upstream nodes v_1, v_2 and v_3.

acquisition can be terminated to minimize the overall cost while satisfying the P_d and P_f constraints.

Based on this sequential detection model, the minimum cumulative log-likelihood ratio required for v_0 to detect the PoI with sufficient accuracy is $S_0 \geq \mathbb{B}$. The IQ threshold I_T can then be directly mapped to $\mathbb{B}$ such that $I_T = \mathbb{B}$. The corresponding IQ provided by each node v_i is hence $q_i = \log \Lambda(b_i)$. As an illustration, the IQ provided by v_7 and v_8 in Figure 4.3 are $q_7 = \log \Lambda(b_7) = 0.4$ and $q_8 = \log \Lambda(b_8) = 0.6$ respectively. The corresponding cumulative log-likelihood ratio at v_3 is $S_3 = \log \Lambda(b_3) + q_7 + q_8 = 1.2 + 0.4 + 0.6 = 2.2$.

4.3.4 Delay Model

To maximize aggregation opportunities and thus minimize energy consumption, a structured aggregation model is adopted in this work. Each node waits for a delay (that linearly decreases with increasing distance/hopcount of the node from the fusion center) before forwarding data to its downstream node. Essentially, an arbitrary node v_i has to wait for data from all the nodes in the subtree rooted at v_i to be transmitted to it, before it can forward data (if any) to its downstream node v_j. For instance, the subtree rooted at v_2 in Figure 4.3 includes v_6, v_{10}, v_{11} and v_{12}. Therefore, v_2 has to wait for data (if any) from all the nodes

in its subtree to be forwarded to it, before it transmits the fused data to its downstream node (and fusion center) v_0.

We denote the **maximum one-hop delay** (inclusive of sleep latency, queuing delay, processing delay, transmission delay and propagation delay) between any two nodes in the network as $\Delta_{\max}$. Then, a node v_i with hopcount h_i has to transmit the aggregated data to its downstream node v_j only at time:

$$t = [(2h_{\max} - h_i)\Delta_{\max} - t_i^A] \mod (h_{\max}\Delta_{\max}), \tag{4.11}$$

where t_i^A is the data arrival time at v_i. This ensures that the fusion center v_0 receives data from all the activated nodes within a delay bound of $h_{\max}\Delta_{\max}$.

4.3.5 Cost Model

Each link between a pair of nodes v_i and v_j is associated with some cost C_{ij}. In the absence of power control, the per-hop link cost C_{ij} is independent of the distance between v_i and v_j, and can be computed as a function of: (i) processing energy required to process and perform data aggregation on a data packet; (ii) transmission energy expended by v_i; and (iii) reception energy expended by neighbors of v_i upon reception of the packet in a wireless medium. One implicit assumption in our cost model is that each data packet is of the same size, regardless of the amount of data that has been fused together from different sensor sources. We explain our assumptions behind this model in Section 4.4.

4.3.6 Problem Formulation

Given the network $G = \{V, E\}$, set of activated nodes V_a, IQ contribution q_a of each activated node $v_a \in V_a$ and IQ threshold I_T, our objective is to design an IQ-aware routing protocol that detects the PoI with an IQ of at least I_T using minimal cost. Formally, we want to find a *subset-τ* Steiner Tree, which is a Steiner Minimum Tree $G_\tau = \{V_\tau, E_\tau\} \subseteq G$ that spans the fusion center v_0 and

all nodes in $V_\tau \subseteq V_a$, such that: (i) aggregated IQ collected from V_τ exceeds I_T; and (ii) total cost of the aggregation tree is minimum among all possible Steiner trees that meet the IQ constraint, i.e.,

$$\min \sum_{e_{ij} \in E_\tau} C_{ij}; \tag{4.12}$$

subject to:

$$\sum_{v_i \in V_\tau} q_i \geq I_T. \tag{4.13}$$

Lemma 3. *The IQ-aware routing problem which finds the least-cost routing path for a given subset $V_\tau \subseteq V_a$ of the activated nodes that satisfies the IQ constraint I_T is NP-hard.*

The proof for NP-hardness is as follows:

Proof. We show that our IQ-aware routing problem is NP-hard by reducing the well-known Steiner tree problem in graphs to it. In the Steiner tree graph problem, the input is a graph $G_s = \{V_s, E_s, W_s\}$ whereby V_s is the set of vertices in the graph, E_s is the set of edges, and W_s is a weighting function on the edges in the graph. Given a set of terminals $S \subseteq V_s$, any tree in G_s that spans S is considered to be a Steiner tree. The cost of the Steiner tree is defined to be the sum of its edge costs (weights). Steiner points which are non-terminal vertices (i.e. $V_s \setminus S$) may be included in the Steiner tree to reduce its cost. The objective of the Steiner tree problem is to find the Steiner Minimum Tree (SMT), which is a least-cost tree spanning all the terminals S.

We map G_s to G by mapping V_s to V and E_s to E. The weighting function W_s is defined such that the weight of an edge in E_s (and hence E) is always of unit cost. This implies that a solution for the Steiner tree problem in graphs is also a solution for the IQ-aware routing problem when the subset of activated nodes $V_\tau \subseteq V_a$ is given. □

In practice, V_τ is unknown and there can be many combinations of V_τ that will satisfy the given IQ constraint. As the least-cost routing for each subset of activated nodes V_τ is a Steiner tree problem, our IQ-aware routing protocol is at least as hard as the NP-hard Steiner tree problem. Although there exists distributed algorithms for the Minimum Spanning Tree (MST) [145] - which provide 2-approximation solutions to the Steiner tree problem - it is difficult to ascertain the set of activated nodes to construct a MST that provides a good approximation for the *subset-τ* Steiner Tree.

In addition to the computational complexity on the fusion center, the acquisition of knowledge on the global network topology and individual IQ contribution of each node incurs high overheads in both communication and computation. Thus, such an approach is impractical in the context of wireless sensor networks which are inherently resource-limited.

In the following sections, we describe a compact and efficient way of representing the network topology and IQ contributions of each node, and then propose a heuristic for solving the NP-hard least-cost IQ-aware routing problem.

4.4 Topology-Aware Histogram-Based Aggregation

We first illustrate our approach using the network topology in Figure 4.3. All the sensor nodes are assumed to be activated, such that $V_a = \{v_1, v_2, ..., v_{11}, v_{12}\}$. The number associated with each node v_i represents its IQ contribution $q_i = \log \Lambda(b_i)$. The cost of data transmission across each link is assumed to be of unit cost, i.e. $C_{ij} = 1 \; \forall e_{ij} \in E$, as: (i) packet size remains constant across each link in our IQ aggregation scheme; and (ii) nodes are assumed to be uniform-randomly distributed in the network.

Table 4.1: Minimum cost aggregation tree for various IQ threshold values in the network topology of Figure 4.3.

IQ threshold I_T	Min-cost aggregation tree	Total IQ S_0
1.0	$\{v_3\}$	1.2
2.0	$\{v_3, v_7, v_8\}$	2.2
	$\{v_2, v_3, v_8\}$	2.2
	$\{v_2, v_3, v_5\}$	2.6
	$\{v_1, v_3, v_8\}$	2.1
	$\{v_1, v_4, v_9\}$	2.1
4.5	$\{v_2, v_3, v_5, v_6, v_8, v_{12}\}$	4.5
	$\{v_1, v_2, v_3, v_4, v_5, v_9\}$	4.7

4.4.1 Motivation

Using a direct (or brute force) approach, each activated sensor $v_i \in V_a$ (with $b_i = 1$) forwards its data to the fusion center v_0, which will then determine if the PoI is present. Such an approach is inefficient as data acquisition from all the activated nodes without any aggregation incurs high communication costs and overloads the fusion center. Furthermore, even if all these data can be obtained by v_0, the optimal least-cost routing tree cannot be found efficiently.

Given a global view of the topology and knowledge of IQ contributions of each sensor node in Figure 4.3, it is possible to compute the minimum cost aggregation tree for various IQ thresholds I_T, as detailed in Table 4.1. If the IQ threshold $I_T = 1.0$, the minimum cost aggregation tree comprises of only $\{v_3\}$. If $I_T = 2.0$, the minimum cost aggregation tree can be $\{v_3, v_7, v_8\}$, $\{v_2, v_3, v_8\}$, $\{v_2, v_3, v_5\}$, $\{v_1, v_3, v_8\}$ or $\{v_1, v_4, v_9\}$. Similarly, if $I_T = 4.5$, then the minimum cost aggregation tree can be $\{v_2, v_3, v_5, v_6, v_8, v_{12}\}$ or $\{v_1, v_2, v_3, v_5, v_5, v_9\}$. However, it is desirable to utilize an efficient and distributed way of computing a minimum cost aggregation tree that meets the IQ constraint I_T.

In the proposed approach, upon the detection of a PoI, activated nodes that are further from the fusion center v_0 initiate the transmission of *hints* towards

it. These hints are *aggregated* by downstream nodes and further propagated towards v_0. The aggregated hint conveyed by an arbitrary node v_i is designed to present useful information about how IQ is distributed in the subtree rooted at v_i, without providing the detailed IQ values of each node and the actual topology of the subtree. For the purpose of scalability, these *hints* are of constant size. The objective is to design a scheme that generates sufficiently useful hints to v_0, so that a minimum cost tree can be constructed in the reverse direction, in a distributed fashion. Our approach is based on the concept of a topology-aware histogram-based (hints) aggregation.

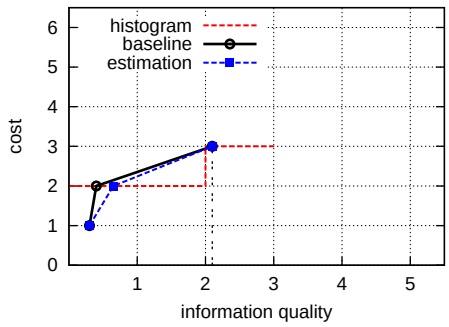
4.4.2 Histogram-Based Representation

In our baseline histogram representation, the y-axis represents the cost (total number of transmissions along routing path, which is proportional to $\sum C_{ij}$) and the x-axis represents the IQ that can be accumulated with the given cost. Depending on the routing path that is taken, different IQ values may be accumulated for a given cost. In this baseline representation, the accumulated IQ is the largest possible for a particular cost. Note that the computation of this *maximum* IQ for a given cost is similar to the original routing problem (which finds the minimum cost of obtaining a particular IQ value), and cannot be computed efficiently for a large network.

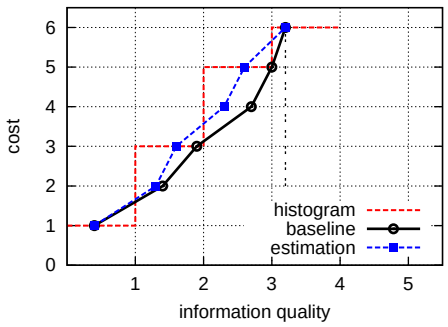
An exhaustive search can be used to compute the maximum aggregated IQ values for each cost c in the small network in Figure 4.3. Table 4.2 lists the maximum IQ $q_i(c)$ and corresponding minimum-cost aggregation tree $M_i(c)$ for each of the upstream nodes of v_0 (i.e., v_1, v_2 and v_3). These values are represented as the solid lines (labeled as *baseline*) in Figure 4.4, which illustrate baseline curves for cost vs maximum IQ. As highlighted previously, such a baseline representation embeds detailed network topology and IQ distribution information, at the cost of excessive computational and communication overheads. To reduce information content and overheads of the representations, quantization levels are

Table 4.2: Baseline of actual IQ $q_i(c)$ and corresponding min-cost aggregation tree $M_i(c)$ per incremental cost c, for each of the upstream nodes of v_0.

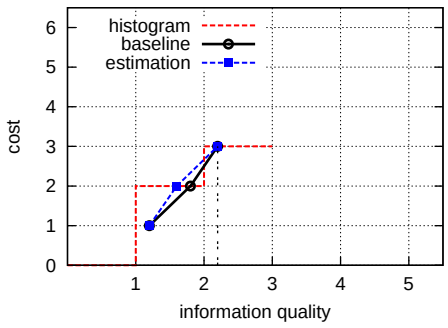
	v_1		v_2		v_3	
cost c	$M_1(c)$	$q_1(c)$	$M_2(c)$	$q_2(c)$	$M_3(c)$	$q_3(c)$
1	$\{v_1\}$	0.3	$\{v_2\}$	0.4	$\{v_3\}$	1.2
2	$\{v_1, v_4\}$	0.4	$\{v_2, v_5\}$	1.4	$\{v_3, v_8\}$	1.8
3	$\{v_1, v_4, v_9\}$	2.1	$\{v_2, v_5, v_6\}$	1.9	$\{v_3, v_7, v_8\}$	2.2
4	-	-	$\{v_2, v_5, v_6, v_{12}\}$	2.7	-	-
5	-	-	$\{v_2, v_5, v_6, v_{11}, v_{12}\}$	3.0	-	-
6	-	-	$\{v_2, v_5, v_6, v_{10}, v_{11}, v_{12}\}$	3.2	-	-



(a) Cost function of v_1.



(b) Cost function of v_2.



(c) Cost function of v_3.

Figure 4.4: Cost functions of subtrees rooted at v_1, v_2 and v_3 (from the perspective of v_0) in Figure 4.3, with IQ threshold $I_T = 5$ and number of discretization levels $\phi = 5$.

introduced.

Quantization

Let ϕ be the number of quantization levels in the baseline representation. This results in ϕ histograms, each of width $\frac{I_T}{\phi}$, where I_T is the IQ threshold required for PoI detection. The range of IQ values represented by the i^{th} block ($i = 1, ..., \phi$) is $(i-1) \cdot \frac{I_T}{\phi}$ to $i \cdot \frac{I_T}{\phi}$ (inclusive).

The number of points within the i^{th} block is denoted as n_i. The corresponding cost c_i for the i^{th} block is $c_i = \sum_{j=1}^{j=i} n_j$. It can be deduced that with a cost of c_{i+1} (where $i < \phi$), the minimum IQ that is obtainable from the network is at least $i \cdot \frac{I_T}{\phi}$. The dotted lines (labeled as *histogram*) in Figure 4.4 illustrate the relationship between the baseline and histogram representations of nodes v_1, v_2 and v_3 from the perspective of v_0 in Figure 4.3.

IQ Estimation

After quantization, the (original) baseline representation of the IQ that is known to an arbitrary node v_e is substituted by a (compact) histogram representation, to be transmitted to its downstream node v_d. We now discuss how v_d estimates the IQ of the subtree rooted at v_e, based on the histogram representation it receives from v_e.

Each of the n_i points in the i^{th} block of the histogram is associated with cost $c_i^k = c_i - n_i + k$, where $1 \leq k \leq n_i$. Assuming that each of these points is uniformly distributed within the block range, the corresponding estimated IQ $\hat{q_e}(c_i^k)$ obtained with a cost of c_i^k (from the perspective of v_d) is given by:

$$\hat{q_e}(c_i^k) = \frac{I_T}{\phi}[(i-1) + \frac{k}{n_i + 1}]. \tag{4.14}$$

Given the IQ q_e obtainable from the upstream node v_e and the maximum IQ q_e^M that can be obtained using the subtree rooted at v_e, the estimated IQ in Equation 4.14 can be further tightened by using q_e and q_e^M as the lower and upper bounds of the histogram. Note that c_i and $\hat{q_e}(c_i^k)$ are undefined $\forall\ i = \lceil q_e^M \cdot \frac{\phi}{I_T} \rceil$,

Table 4.3: Estimated and actual IQ gain per incremental cost c from perspective of v_0.

Cost c	v_1		v_2		v_3	
	$q_1(c)$	$\hat{q_1}(c)$	$q_2(c)$	$\hat{q_2}(c)$	$q_3(c)$	$\hat{q_3}(c)$
1	0.3	0.3	0.4	0.4	1.2	1.2
2	0.4	0.65	1.4	1.3	1.8	1.6
3	2.1	2.1	1.9	1.6	2.2	2.2
4	-	-	2.7	2.3	-	-
5	-	-	3	2.6	-	-
6	-	-	3.2	3.2	-	-

$k \in \mathbb{Z}^+$ as these are not regions of interest.

The dotted lines with points (labeled *estimation*) in Figure 4.4 plot the values of estimated IQ $\hat{q_e}(c)$ for each additional edge from the subtree rooted at v_e, from the perspective of its downstream node v_d. The *estimation* plot can be interpreted in this way: For a cost of c, it is likely that at least an IQ of $\hat{q_e}(c)$ can be obtained. Table 4.3 compares the baseline $q_e(c)$ and estimated $\hat{q_e}(c)$ values for each of the subtrees rooted at the upstream nodes of v_0 in Figure 4.3. Note that $q_e(c)$ can be larger or smaller than $\hat{q_e}(c)$.

Finally, we discuss the selection of the parameter ϕ. In our IQ estimations, each of the n_i points in the i^{th} block is assumed to be uniformly distributed; the corresponding IQ estimations for each of the points is also uniformly distributed. If this assumption is valid, then the value of ϕ can be small. However, if IQ values vary significantly among nodes, then a larger ϕ value is required to increase the accuracy of the piecewise linear approximation. We can now describe our proposed IQ-aware routing protocol in the following section.

4.5 IQ-Aware Routing Protocol

In an event-driven wireless sensor network, data generated by a sensor provides information about the likelihood that a PoI has occurred. Section 4.3 describes how this data is mapped to the information quality q_a provided by an activated node $v_a \in V_a$. To minimize the cost of data transmission, activated nodes first generate hints that are aggregated towards the fusion center v_0 so that a minimum cost detection tree can be constructed. The IQ threshold I_T required to detect the PoI is assumed to be known.

Upon activation, each node v_j transmits hints to its downstream node v_i in the form of a quadruple comprising:

1. information quality q_j of v_j;
2. maximum information quality q_j^M that can be obtained using the subtree rooted at v_j;
3. maximum cost C_j^M of the subtree rooted at v_j; and
4. histogram $\{n_j^1, n_j^2, ..., n_j^\phi\}$ representing the topology-aware IQ obtainable using the subtree rooted at v_j.

4.5.1 Initialization

A distance-based aggregation tree is constructed using a shortest-path algorithm during network initialization. To maximize aggregation opportunities, data generated by an activated node is transmitted only after a delay that linearly decreases with increasing distance of the node from v_0. As an activated leaf node (without activated upstream nodes) has only one data point, building the histogram is trivial using the method in Section 4.4.2. For example, the quadruples transmitted by v_{10}, v_{11} and v_{12} to their (common) downstream node v_6 are $\{0.2, 0.2, 1, \{1, \emptyset, \emptyset, \emptyset, \emptyset\}\}$, $\{0.3, 0.3, 1, \{1, \emptyset, \emptyset, \emptyset, \emptyset\}\}$ and $\{0.8, 0.8, 1, \{1, \emptyset, \emptyset, \emptyset, \emptyset\}\}$ respectively. Non-leaf nodes can have multiple upstream nodes.

4.5.2 Aggregation and Update

The histogram at node v_i is updated in three main phases.

- In Phase 1, v_i estimates the IQ $\hat{q_j}(c_j)$ that can be obtained for each cost $1 \leq c_j \leq C_j^M$, using the subtree rooted at each upstream node $v_j \in V_i^u$.
- Phase 2 is triggered if v_i has multiple (activated) upstream nodes. A greedy heuristic is used to approximate the maximum IQ obtainable for each given cost $1 \leq c_i \leq \sum C_j^M$.
- In Phase 3, v_i incorporates its cost and IQ q_i into the IQ estimations obtained in earlier phase(s), and translates these estimations back into a (new) histogram for transmission to its downstream node.

We now detail each of these phases, using numerical examples from the topology in Figure 4.3, with $I_T = 5$ and $\phi = 5$.

Phase 1

Estimation of the IQ $\hat{q_j}(c_j)$ for the subtree rooted at each upstream node v_j is done by utilizing the histogram which is part of the quadruple transmitted from v_j to v_i. Considering the subtree rooted at v_3, the quadruples transmitted by v_7 and v_8 to v_3 are $\{0.4, 0.4, 1, \{1, \emptyset, \emptyset, \emptyset, \emptyset\}\}$ and $\{0.6, 0.6, 1, \{1, \emptyset, \emptyset, \emptyset, \emptyset\}\}$ respectively. v_3 then estimates the information quality $\hat{q_7}(c_7)$ $\forall 1 \leq c_7 \leq C_7^M$ and $\hat{q_8}(c_8)$ $\forall 1 \leq c_8 \leq C_8^M$ using Equation 4.14. Since v_7 and v_8 are both activated leaf nodes, v_3 can easily and accurately estimate $\hat{q_7}(1) = 0.4$ and $\hat{q_8}(1) = 0.6$.

Phase 2

Since v_3 has multiple upstream nodes, the second phase of the algorithm is invoked. From a global perspective, it is trivial to see that with a cost of 1, only v_3 is included in the routing path. Similarly, with a cost of 3, all the three nodes (v_3, v_7 and v_8) in the subtree rooted at v_3 are included. However, with a cost of

Algorithm 3 IQ Approximation Algorithm in Phase 2

1: **Input:** I_T, $\hat{q_j}(c_j)$ $\forall v_j \in V_i^u$, $1 \leq c_j \leq C_j^M$
2: **Variable:** $I_{total} = 0$, $\bar{c_j} = 0$ $\forall v_j$
3: **Output:** $\hat{q_i}(c_i)$ $1 \leq c_i \leq C_i^M$
4: **while** $\sum \bar{c_j} < \sum C_j^M$ OR $I_{total} < I_T$ **do**
5: $k \leftarrow \underset{j}{\text{argmax}} \, [\hat{q_j}(\bar{c_j} + 1) - \hat{q_j}(\bar{c_j})]$
6: $\bar{c_k} \leftarrow \bar{c_k} + 1$
7: $I_{total} \leftarrow I_{total} + [\hat{q_j}(\bar{c_j} + 1) - \hat{q_j}(\bar{c_j})]$
8: $\hat{q_i}(\sum \bar{c_j}) \leftarrow I_{total}$
9: **end while**

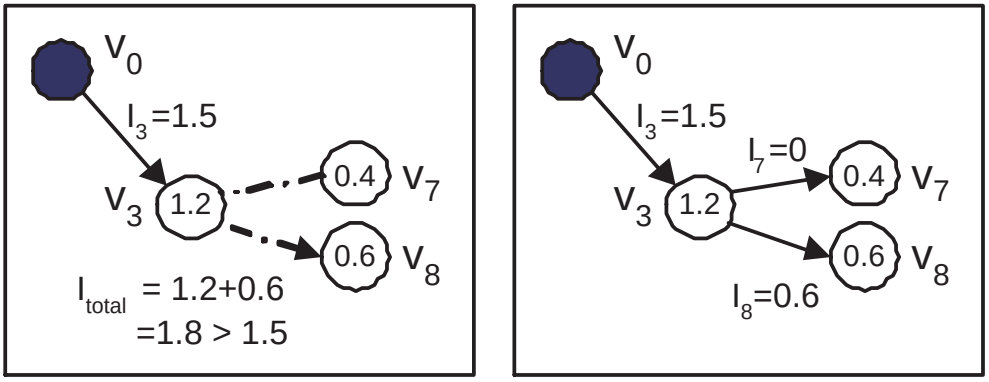
2, either v_7 or v_8 is included, with the latter yielding a higher cumulative IQ. The general complexity of computing the highest IQ for each cost $1 \leq c_i \leq \sum C_j^M$ using an exhaustive brute-force search is $\prod_{v_j \in V_i^u} C_j^M$.

We approximate the maximum IQ for each cost using a greedy heuristic that significantly reduces computational complexity while maintaining reasonable IQ accuracy. Let I_{total} be the estimated cumulative IQ of all sensors that are included in the minimum cost tree M_i (initially empty). Recall that v_i has computed $\hat{q_j}(c_j)$ for each incremental cost $1 \leq c_j \leq C_j^M$ along each subtree rooted at $v_j \in V_i^u$ in Phase 1. Let $\bar{c_j}$ denote the current cost of the subtree rooted at v_j, that has been included in M_i; initially, $\bar{c_j} = 0$ $\forall v_j$.

At each iterative step, the estimated IQ $\hat{q_k}(\bar{c_k}+1)$ that provides the maximum IQ gain to I_{total} is included in M_i. The IQ gain is computed by $\hat{q_k}(\bar{c_k}+1) - \hat{q_k}(\bar{c_k})$. This process repeats until: (i) all the subtrees rooted at the upstream nodes have been added to M_i; or (ii) I_{total} exceeds the IQ threshold I_T. Based on the subtree rooted at v_3 (excluding itself) in Figure 4.3, the estimated maximum IQ values for the different costs using this greedy heuristic, are $\hat{q_3}(1) = 0.6$ and $\hat{q_3}(2) = 0.6 + 0.4 = 1.0$. Algorithm 3 summarizes the IQ approximation procedure that is executed by v_i in Phase 2.

Algorithm 4 Histogram Construction Algorithm in Phase 3

1: Input: $\hat{q_i}(c_i)$ $1 \leq c_i \leq C_i^M$
2: Variable: $j = 0$
3: Output: $\{n_i^1, n_i^2, ..., n_i^\phi\}$
4: $n_i^k \leftarrow 0 \qquad 1 \leq k \leq \phi$
5: **for** $k = 1$ to C_i^M **do**
6: $\quad j \leftarrow \lceil \hat{q_i}(c_k) \cdot \frac{\phi}{I_T} \rceil$
7: $\quad n_i^j \leftarrow n_i^j + 1$
8: **end for**



(a) v_3 includes v_8 into the pruned tree and updates I_{total}.

(b) v_3 assigns the appropriate IQ thresholds to its upstream nodes.

Figure 4.5: Sequence of pruning activities for subtree rooted at v_3.

Phase 3

In the final step of the algorithm, the estimated maximum IQ obtained for the subtree rooted at v_i is updated to include its own cost and IQ q_i. The values of $\hat{q_3}(c_i)$, where $1 \leq c_i \leq C_i^M$ are updated such that $\hat{q_3}(1) = 1.2$, $\hat{q_3}(2) = 1.2 + 0.6 = 1.8$ and $\hat{q_3}(3) = 1.2 + 1.0 = 2.2$. Based on this updated set of IQ estimations, a new quantized histogram is constructed and forwarded to the downstream node v_0. Algorithm 4 summarizes the histogram construction procedure in Phase 3.

4.5.3 Pruning

The pruning phase commences after v_0 receives data from its upstream nodes $v_j \in V_0^u$. Its objective is to prune off as many nodes as possible from the initial distance-based aggregation tree, such that: (i) IQ constraint I_T is still satisfied;

and (ii) total transmission cost of collecting data from the resulting pruned tree is minimized. Hence, v_0 has to allocate an IQ threshold I_j (with corresponding estimated cost $\hat{c_j} \leq C_j^M$) to each upstream node v_j such that: (i) $\sum I_j \geq I_T$; (ii) $\hat{q_j}(\hat{c_j}) \geq I_j$; and (iii) $\sum \hat{c_j}$ is minimized.

The pruning algorithm adopts a greedy approach similar to that in Phase 2 of the data aggregation algorithm. Based on previous computations, each node v_i has the estimated maximum IQ $\hat{q_j}(c_j)$ of all its upstream nodes $v_j \in V_i^u$ for each cost $1 \leq c_j \leq C_j^M$. The pruned routing tree is initially empty with total IQ $I_{total} = 0$. v_i iteratively includes into its pruned tree, the value of $\hat{q_j}(c_j)$ that provides maximum IQ increment. This process repeats at v_i until: (i) subtree rooted at each upstream node v_j is included in the pruned tree; or (ii) aggregated IQ I_{total} of the pruned tree exceeds the IQ threshold I_i at v_i. The expected output of the pruning algorithm at v_i is the assignment of I_j to each upstream node v_j. The pruning algorithm is executed recursively at each upstream node v_j along the initial aggregation tree using I_j.

An activated node v_j with assigned IQ threshold $I_j = 0$ is considered to be **pruned** off and not required to be part of the resulting pruned tree. Pruned nodes suppress their data for a time epoch before resuming the forwarding of data towards v_0. The temporary suppression of data enables the aggregation routing path to be adaptive towards dynamic changes in the network and PoI, while reducing transmission costs.

Assuming that v_0 has assigned $I_3 = 1.5$ in Figure 4.3, we look at how v_3 assigns I_7 and I_8 to its upstream nodes v_7 and v_8. Since $\hat{q_8}(1) > \hat{q_7}(1)$, the former is included into the pruned tree of v_3 and $I_{total} = q_3 + \hat{q_8}(1) = 1.2 + 0.6 = 1.8 > 1.5 = I_3$, as in Figure 4.5(a). Since the IQ threshold at v_3 is met, the pruning algorithm at v_3 terminates with $I_7 = 0$ and $I_8 = 0.6$, as shown in Figure 4.5(b). v_7 is temporarily pruned off from the routing tree to minimize costs as $I_7 = 0$. Pruning terminates here as v_7 and v_8 are activated leaf nodes.

4.5.4 Discussion

The advantage of the proposed aggregation scheme is that while the fusion center sees a highly aggregated summary of how IQ is distributed in the sensor network, the accuracy improves in the pruning process when specific subtrees are selected. Hence, one can think of the aggregation process as building a distributed structure that allows IQ distribution information to be selective refined as needed by the pruning process.

Finally, each sensor node has a low but non-zero probability of false positive event detection. If the network is sufficiently large, aggregation of a large number of sensors that falsely detect an event may be sufficient to trigger event detection. Such false alarms can be handled by the fusion center using simple heuristics: Since sensors with false positives are randomly distributed, the cost for event detection is very large as compared to that of normal event detection; hence, a simple cost threshold may be used to suppress such false positives.

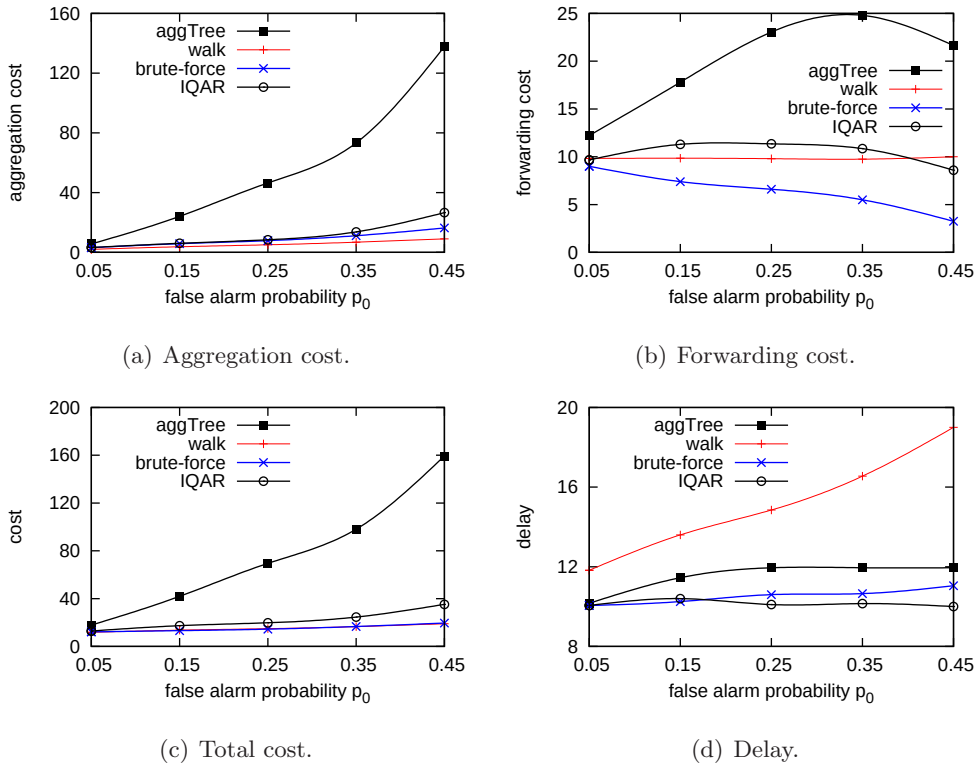
4.6 Performance Evaluation

We evaluate the performance of IQAR in Qualnet 4.0 [146] and compare against the following routing protocols:

1. *aggTree*: Distance-based aggregation tree that collects data from all activated sensors, to be processed at v_0.

2. *walk*: IQ-aware routing protocol that routes data greedily towards next hop with highest IQ. When no IQ can be further gained from neighbors of the transmitter, or when aggregated data has sufficient IQ, data is routed back to v_0 using a shortest path algorithm. Routing process is initiated from node with highest global IQ among all activated nodes, and there is only one ongoing transmission (and routing path) at any time.

3. *brute-force*: IQ-aware routing protocol that is similar in operations to



(a) Aggregation cost.

(b) Forwarding cost.

(c) Total cost.

(d) Delay.

Figure 4.6: Performance with increasing per-sample false alarm probability p_0.

IQAR, but uses an exhaustive brute-force search to compute the maximum IQ for each cost function during data aggregation and pruning.

The fusion center v_0 is located near the bottom left hand corner of the terrain of size $\{100m \times 100m\}$. All the other sensors are uniform-randomly distributed in the network. The performance result illustrated is averaged over the sensing interval (1 second), and 20 seed runs. The target detection and false alarm probabilities are $P_d = 0.9$ and $P_f = 0.001$ respectively, yielding a IQ threshold of $I_T = \log \frac{P_d}{P_f}$. The transmission range of each node is approximately 8 meters.

4.6.1 Varying Local Information Quality

The per-sample false alarm probability p_0 is varied in Figure 4.6, which leads to: (i) increase in number of activated nodes and detection region; and (ii) decrease in local IQ of each node. The PoI occurs at a fixed location $\{80m \times 80m\}$ and

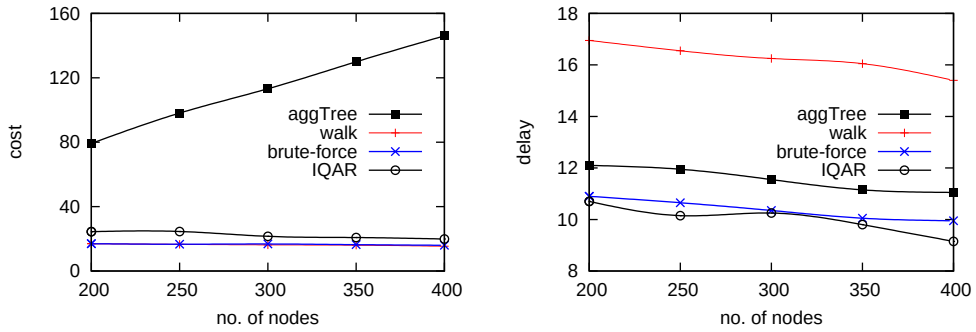
the network has 250 nodes.

The aggregation cost in Figure 4.6(a) measures number of transmissions involving aggregated data, and is highly correlated to the number of nodes with aggregated data. Due to the lowered per-node IQ as p_0 increases, more sensory data have to be aggregated to meet the IQ threshold I_T, resulting in increased aggregation cost for all protocols. *aggTree* incurs the highest aggregation cost as it fuses data from all the activated nodes. In contrast, the three IQ-aware protocols - *walk*, *brute-force* and *IQAR* - aggregate data from only a subset of the activated nodes and incur less aggregation costs.

The forwarding cost in Figure 4.6(b) measures number of transmissions required to forward data from the last aggregated node to v_0, and is dependent on activated node locations as well as PoI location. The enlarged detection region resulting from the increase in p_0 leads to a corresponding increase in forwarding cost, especially for *aggTree* as it collects data from all the activated nodes. *IQAR* incurs higher costs than *brute-force* as the former adopts a greedy approach that may not yield the best routing path for a given IQ threshold. The gradual decrease in forwarding cost observed for all the protocols as p_0 increases is due to the activation of *more* sensor nodes that are nearer to v_0, which decreases the distance between v_0 and first aggregated node along each routing path.

Figure 4.6(c) illustrates the total cost of aggregating and forwarding data packets to v_0. Since aggregation cost dominates over forwarding cost in a network with a small network diameter, the total cost has a similar trend to aggregation cost.

The delay incurred in Figure 4.6(d) is measured in terms of number of (sequential) transmissions. Despite the low transmission cost incurred by *walk*, it incurs the highest delay as transmissions occur sequentially along a single path, where aggregation occurs strictly before forwarding. Due to the presence of multiple paths in *brute-force* and *IQAR*, multiple aggregation and forwarding of data can take place simultaneously, thus reducing the overall delays.



(a) Total cost. (b) Delay

Figure 4.7: Performance with increasing network density.

Note that the cost incurred by *aggTree* provides an upper bound for IQAR, as the former collects data from all the sensors in the network. Although IQAR does not make use of global information as like in *walk*, it is able to achieve comparable performance to the latter.

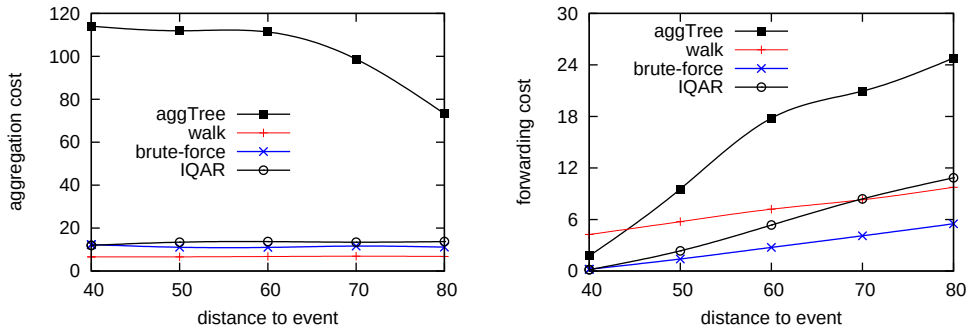
4.6.2 Varying Network Density

In Figure 4.7, network size increases from 200 to 400 nodes and PoI is located at $\{80m \times 80m\}$ with $p_0 = 0.35$. Due to the increase in network size (and density), the number of activated nodes increase. The total cost incurred by *aggTree* increases correspondingly in Figure 4.7(a) as it collects data from all activated nodes. The remaining three IQ-aware protocols do not collect data from all the activated nodes and can achieve lower costs. The excessive delays incurred by *walk* in Figure 4.7(b) highlights the caveat of having only a single routing path that limits parallelism of data aggregation and forwarding.

4.6.3 Varying Distance between Event (PoI) and Fusion Center

We vary the distance between the PoI and fusion center v_0 in a network of 250 nodes with $p_0 = 0.35$ in Figure 4.8. The x/y coordinates of the PoI are varied from $40m$ to $80m$[2].

[2]A distance of 60 to the PoI implies that the PoI is located at $\{60m \times 60m\}$.



(a) Aggregation cost. (b) Forwarding cost.

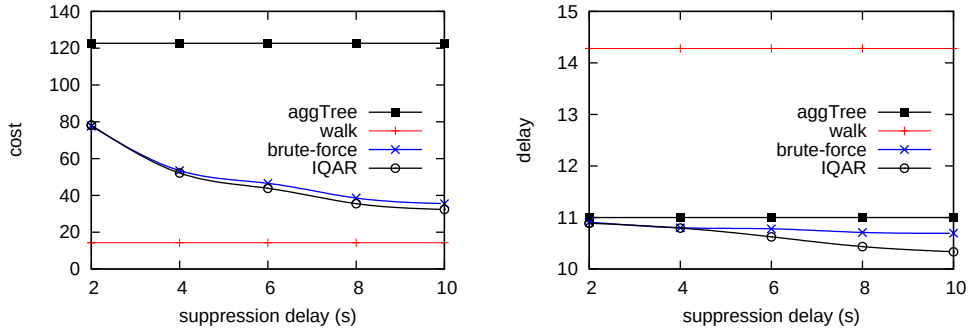
Figure 4.8: Performance with increasing distance to event (PoI).

As distance to event increases, it is located nearer to edges of the terrain, leading to decrease in number of activated nodes. Consequently, the number of fused nodes and aggregation cost using *aggTree* decreases in Figure 4.8(a). The aggregation costs remain relatively constant for the IQ-aware protocols as they aggregate data from the minimum possible number of nodes to satisfy the IQ threshold. Forwarding cost generally increases with increasing distance to PoI in Figure 4.8(b), as more transmissions are required to forward data from activated sensors to v_0. *IQAR* uses a greedy heuristic to estimate IQ contributions of nodes; hence it incurs higher forwarding cost than *brute-force*. Despite the low aggregation cost incurred by *walk*, it incurs relatively higher forwarding costs as the activated sensor with the highest global IQ may be further away from v_0 than other activated nodes.

4.6.4 Varying Suppression Interval

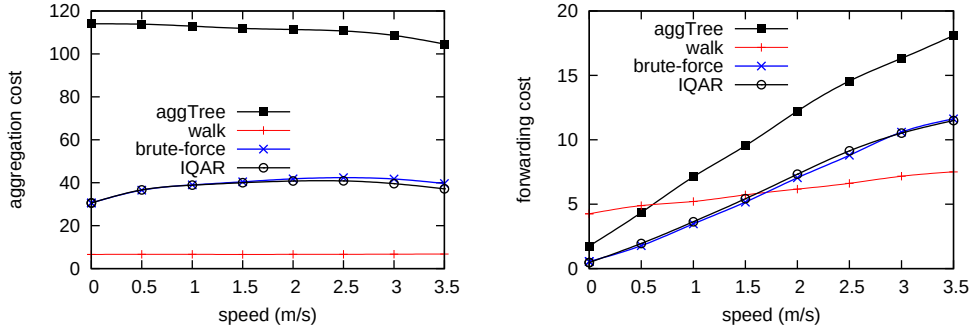
In the above scenarios, the PoI is statically located throughout the monitoring period. Both *brute-force* and *IQAR* can achieve significant cost and delay savings over *aggTree* as they aggregate data from only a subset of activated nodes to satisfy the IQ threshold. Data from the remaining activated nodes are suppressed for a **suppression interval** to reduce transmission costs[3]. One main concern

[3]With a suppression interval of x seconds, an activated node suppresses its data for at least x seconds after its last transmission.



(a) Total cost. (b) Delay

Figure 4.9: Performance with varying suppression interval (delay).



(a) Aggregation cost. (b) Forwarding cost.

Figure 4.10: Performance with varying event (PoI) mobility.

with such protocols (which utilize data suppression to reduce costs) is whether a mobile PoI can be detected with sufficient IQ.

In Figure 4.9, the speed of the PoI is fixed at 2.5 ms^{-1} in a network of 250 nodes with $p_0 = 0.35$. As suppression interval increases from 2s to 10s, the amount of aggregated data decreases as nodes are suppressed for longer periods of time. Subsequently, there is a decrease in total cost and delay of *brute-force* and *IQAR* in Figures 4.9(a) and 4.9(b). However, it should be noted that due to the suppression of data, the detection accuracies achieved by these two protocols deteriorate by 5% to 10% with mobile PoIs. The total cost and delay incurred by *aggTree* and *walk* remain constant as they do not suppress data.

4.6.5 Varying Event Mobility

In Figure 4.10, the PoI moves with varying speeds from 0 ms^{-1} to 3.5 ms^{-1} diagonally across the network with 250 nodes and $p_0 = 0.35$. The suppression interval is fixed at 5 seconds.

As event mobility increases, forwarding cost in Figure 4.10(b) increases as the PoI is located increasingly further away from v_0. As the PoI also exits the suppression region more quickly with higher event mobility, more sensors are activated, leading to the increased aggregation cost incurred by *brute-force* and *IQAR* in Figure 4.10(a). Since the PoI moves diagonally across the network, it is much closer to the network edge at higher mobilities, which limits the number of activated nodes. Hence, the aggregation costs incurred by *aggTree*, *IQAR* and *brute-force* drop slightly when speed exceeds 2 ms^{-1}.

4.7 Summary

In this work, we propose IQAR - an Information Quality Aware Routing protocol for event-driven sensor networks. IQAR considers the individual IQ contribution of each sensory data, and collects only sufficient data for a phenomenon of interest (PoI) to be detected reliably. Redundant data is suppressed for a time interval to reduce traffic load and alleviate medium access contention. This allows IQAR to achieve significant energy and delay savings while maintaining information quality in event detection.

Chapter 5

Information Quality Delay Efficient Aggregation

Chapter 4 highlights the effectiveness of data aggregation in reducing energy expenditure in energy constrained wireless sensor networks. However, there exists an inevitable energy-delay tradeoff with the use of such in-network processing techniques, as aggregation opportunities are increased (resulting in greater energy savings) at the cost of longer delays. This chapter focuses on how fast and energy efficient data aggregation can be achieved in an event driven wireless sensor network. We present IQDEA - an *I*nformation *Q*uality *D*elay *E*fficient *A*ggregation scheme which is able to strike a good balance between detection latency and energy efficiency in these networks.

5.1 The Energy-Delay Tradeoff

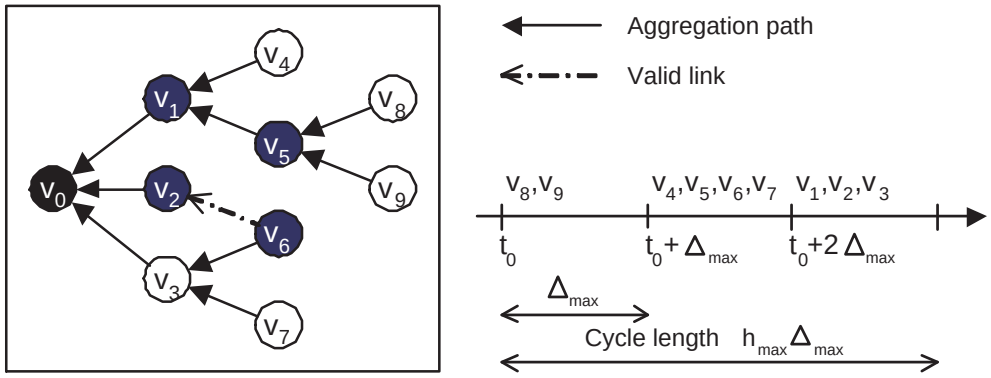
By combining spatially and/or temporally correlated data from multiple sources into a single packet before forwarding it to the fusion center, data aggregation [123] [124] [91] [125] has the potential to bring about substantial reductions in traffic volumes and thus communication costs. In event driven sensor networks, aggregation can also help to alleviate the detrimental effects of data implosion

and redundancy upon the occurrence of Phenomena of Interest (PoI). Generally, by waiting for a longer period of time before forwarding data towards the fusion center, there is a larger opportunity for data aggregation to take place. A natural tradeoff therefore exists between increasing aggregation opportunity at each node and reducing detection delay at the fusion center.

On one hand, there should be a minimum delay between local event detection and data transmission at each node, so that PoI detection delay at the fusion center can be minimized. On the other hand, without waiting or exploiting some form of coordination, data aggregation becomes less likely and overall number of transmissions increases. The task of introducing some structure in the waiting period is a challenging problem since the PoI can occur anywhere. If the worst scenario (that data has to be acquired from all the nodes in the network and maximum delay is incurred at each hop along the routing path) is not assumed, it is difficult to quantify the amount of data aggregation that is actually possible.

We highlight the varying energy-delay tradeoffs experienced by different data aggregation schemes using the simple network topology in Figure 5.1(a), which comprises of the fusion center v_0 and sensor nodes $v_1, v_2, ..., v_9$. The hopcount of an arbitrary node v_i is denoted as h_i, and the network diameter is denoted as $h_{\max}$. The hopcounts of the sensor nodes are $h_1 = h_2 = h_3 = 1$, $h_4 = h_5 = h_6 = h_7 = 2$ and $h_8 = h_9 = h_{\max} = 3$.

With the use of structured schemes such as aggregation trees, data transmissions are scheduled within periodic cycles (of length $h_{\max}\Delta_{\max}$), where $h_{\max}$ is the network diameter and $\Delta_{\max}$ is the maximum one hop delay across the links. To maximize aggregation opportunities, transmission times of nodes are staggered such that each node is scheduled for transmission only after it has received data from all its upstream (children) nodes. We reference the start time of a cycle by t_0. As illustrated in Figure 5.1(b), nodes v_8, v_9 with hopcount $h_{\max}$ are scheduled to transmit at time t_0; v_4, v_5, v_6, v_7 with hopcount $h_{\max} - 1$ are scheduled to transmit at time $t_0 + \Delta_{\max}$; and v_1, v_2, v_3 with hopcount $h_{\max} - 2$



(a) Aggregation paths with set of activated nodes $V_a = \{v_1, v_2, v_5, v_6\}$.

(b) Timeline of scheduled transmissions in a structured aggregation tree.

Figure 5.1: Illustration of the delays incurred by a structured aggregation tree.

are scheduled to transmit at time $t_0 + 2\Delta_{\max}$.

We assume that the set of activated nodes in Figure 5.1(a) is $V_a = \{v_1, v_2, v_5, v_6\}$ when a PoI occurs at time t_0. Then, using structured aggregation, the delay required to acquire data from all the activated nodes is $h_{\max}\Delta_{\max} = 3\Delta_{\max}$ with a total cost of 5 transmissions. In structureless schemes such as opportunistic aggregation, each node forwards data to its downstream (parent) node immediately upon data arrival. As the maximum hopcount of the activated nodes in Figure 5.1(a) is 2, data from all the nodes in V_a can be received at v_0 within a shorter delay of $2\Delta_{\max}$[1]. However, this comes at higher cost of 6 data transmissions.

5.2 The Case for Energy *and* Delay Efficiency

Despite the existence of energy-delay tradeoffs, many existing protocols for energy constrained networks tend to optimize only energy efficiency, and overlook the significance of the end-to-end delays. This is highlighted in Gu et al [36]:

> "... many energy management protocols ... although are very effective to minimize energy consumption in the network, they rarely

[1] In the general case, the expected delay to acquire data from all the activated nodes using structureless (opportunistic) aggregation is given by $\frac{h_{\max}\Delta_{\max}}{2}$.

consider the impact of resulting node working schedules on communication delay."

This work asserts that while end-to-end delay bounds are generally sufficient for many systems, there are also many other event driven applications whereby delay efficiency can vastly improve overall system performance. For example, in intrusion detection systems or mission critical applications (such as tsunami or fire detection systems), it is evident that the faster the PoI can be detected, the earlier search-and-rescue operations can be deployed, leading to significant reduction in casualties and infrastructural damages.

Our work is complementary to efforts that aim to minimize energy consumption in day-to-day operations of the network - such as duty cycling mechanisms which are effective in prolonging network lifetime in periodic monitoring network applications. However, there exist shortcomings in existing energy efficient protocols that lead to suboptimal delay performance in the system when a PoI occurs, which we highlight as follows:

S_1: Many existing aggregation schemes [35] [70] [71] [79] [80] [81] [82] [83] [84] [85] [86] [87] [88] [91] [92] [93] [94] that aim to minimize energy consumption assume that: (i) data is required from all nodes in the network for PoI detection; and (ii) maximum per-hop delay is incurred at each hop.

S_2: Information quality is not exploited to minimize the amount of data required for PoI detection at the fusion center.

S_3: Duty cycling schemes and routing protocols are designed independently.

[S_1] Aggregation schemes that minimize energy consumption by increasing aggregation opportunities are often structured in nature. In these schemes, the aggregation schedule of each node is pre-defined during network initialization; hence they are more suitable for periodic monitoring applications in which every node has to report its data back to the fusion center at regular time intervals.

In event driven sensor networks, only a subset of nodes may be activated and have sensory data to forward to the fusion center upon the occurrence of the PoI. However, as the locations of the PoI and activated nodes cannot be known *a priori*, the aggregation schedules adopted by these networks assume the worst case scenario whereby: (i) data is required from all the nodes in the network for reliable and accurate PoI detection; and (ii) maximum per-hop delay is incurred at each hop. Taking the earlier numerical example in Figure 5.1, we observe that in the optimal case, the smallest possible cost of 5 data transmissions and shortest possible delay of $2\Delta_{\max}$ to acquire data from all the activated nodes can be achieved if: (i) v_5, v_6 transmit at time t_0; and (ii) v_1, v_2, v_3 transmit at time $t_0 + \Delta_{\max}$. In addition, a smaller cost of 4 transmissions is incurred if forwarding nodes are dynamically selected based on real-time aggregation opportunities, which enables data from v_6 to be aggregated and forwarded by v_2 instead of being forwarded by v_3 without aggregation. However, these require all nodes in the network to have *a priori* knowledge of activated sensor locations.

[S_2] We further note that in many event driven sensor network applications: (i) data generated by each activated sensor node has varying information quality (IQ); and (ii) PoI can be detected with sufficient accuracy and reliability using data from only a subset of the activated nodes (Chapter 4). Through exploitation of IQ awareness, the amount of sensory data that has to be forwarded to the fusion center can be reduced as data acquisition can terminate as soon as the aggregated data satisfies a pre-determined IQ threshold. This minimizes energy consumption as well as reduces the delay required for PoI detection. For example, if only data from activated nodes v_1, v_2 and v_5 in Figure 5.1(a) are required for PoI detection, then total cost and delay can be reduced to 3 data transmissions and $2\Delta_{\max}$ respectively.

[S_3] Although existing duty cycling medium access schemes and routing protocols are generally interoperable with each other, most of them are designed independently of the other without any cross layer interactions. Consequently,

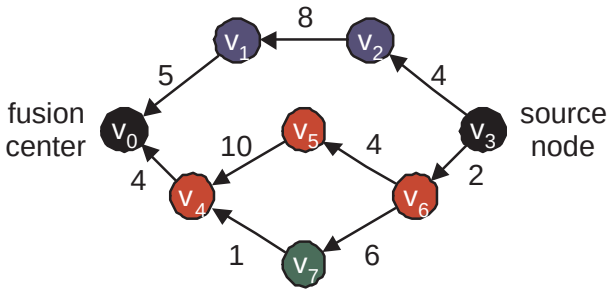


Figure 5.2: Network with duty cycling, where the weight on each edge represents the expected sleep latency (in units) incurred in transmitting along that particular link.

end-to-end delays incurred during data forwarding are not optimized. We illustrate the inefficiency of delay performance arising from the independence between duty cycling and routing designs, using the topology in Figure 5.2, where v_0 is the fusion center and v_3 is the source node. The weight on each edge represents the expected sleep latency (in units) incurred by the upstream node in transmitting along that particular link. It is trivial to see that the least cost path (measured in terms of number of hops, and which does not consider the sleep latencies incurred by duty cycling) is $v_3 \rightarrow v_2 \rightarrow v_1 \rightarrow v_0$, thereby incurring a cost of 3 hops and end-to-end delay of 4+8+5=17 units. Conversely, the least delay path is $v_3 \rightarrow v_6 \rightarrow v_7 \rightarrow v_4 \rightarrow v_0$, thereby incurring a slightly increased cost of 4 hops and significantly reduced end-to-end delay of 2+6+1+4=13 units. However, the computation of these two paths (least cost and least delay) requires knowledge of the global network topology and global duty cycling schedule, which is costly and therefore impractical, especially in large scale multihop networks.

Based on current literature, existing data aggregation schemes can be classified as structured or structureless approaches. Figure 5.3 shows that 'Structured aggregation' schemes tend to minimize energy consumption at the expense of long PoI detection delays, while structureless schemes such as 'Opportunistic aggregation' tend to minimize PoI detection delays at the expense of higher energy consumption. A good data aggregation scheme is therefore one whose

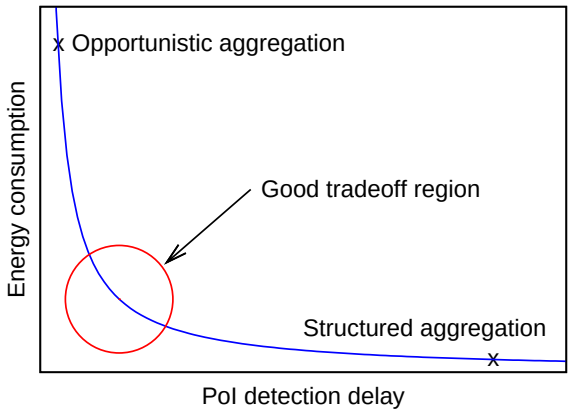


Figure 5.3: Energy consumption vs PoI detection delay for different classes of aggregation schemes.

performance falls within a 'Good tradeoff region' between energy consumption and PoI detection delay. The main focus of this work is thus to introduce a novel data aggregation scheme that provides good energy-delay tradeoffs, by addressing the three key shortcomings of existing protocols as described earlier. The proposed IQDEA scheme is a distributed data aggregation algorithm for event driven sensor networks that provides a good balance between energy and delay efficiency, while taking into account application-layer IQ requirements.

5.3 Preliminaries

In this section, we describe our system model, define the minimum **PoI detection delay** D_P in an event driven sensor network, and formulate the problem statement.

5.3.1 System Model

The system model is similar to that described in Chapter 4.3. The network is represented as a graph $G = \{V, E\}$, where $V = \{v_0, v_1, ..., v_n\}$ is the set of n sensor nodes and fusion center v_0; and E denotes the set of edges. An edge $e_{ij} \in E$ represents the existence of a communication link between an arbitrary

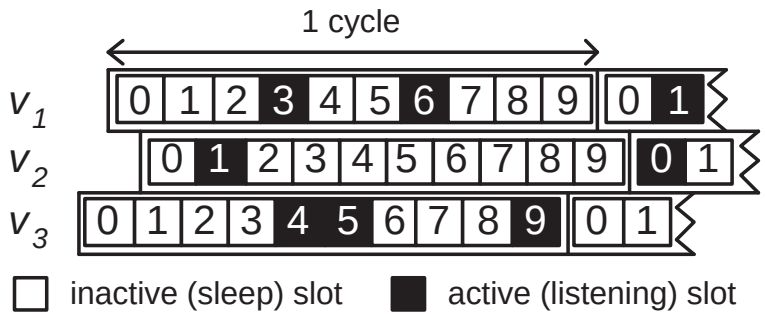


Figure 5.4: Wakeup schedules of v_1, v_2 and v_3 with $\alpha_1 = 2$, $\alpha_2 = 1$ and $\alpha_3 = 3$, in a cycle with $n_c = 10$ slots.

pair of nodes $v_i, v_j \in V$ when they are both active (awake) at the same time. IQDEA does not require a specific MAC or routing protocol; it is interoperable with any (possibly duty cycled) MAC protocol that provides expected end-to-end delay information from each neighboring node to the fusion center and any routing protocol that uses a distance-based metric (such as hopcount or ETX). However, for demonstrability, we describe how IQDEA is integrated with a duty cycled MAC protocol and a hopcount-based routing protocol.

The duty cycling wakeup schedule adopts a generic asynchronous slot model similar to that used in X-MAC [58] and A^2-MAC (Chapter 3). The cycle is composed of n_c slots, each of which is of length τ. The corresponding cycle length is given by $n_c\tau$. Each slot in the cycle can be an active (listening) slot or inactive (sleep) slot. The number of *active* listening slots per cycle that is associated with each node v_i is denoted as α_i, where $1 \leq \alpha_i \leq n_c$. As such, each of the n_c slots within a cycle has $\frac{\alpha_i}{n_c}$ probability of being an active slot. Figure 5.4 illustrates the wakeup schedules of 3 nodes v_1, v_2 and v_3 with $\alpha_1 = 2$, $\alpha_2 = 1$ and $\alpha_3 = 3$ respectively, in a cycle with $n_c = 10$ slots. Note that each node in the network may have a different active probability.

The probing mechanism in A^2-MAC is integrated into the medium access control operation to guarantee communication between any pair of nodes v_i and v_j, as illustrated in Figure 5.5. When a packet arrives at an arbitrary node v_i,

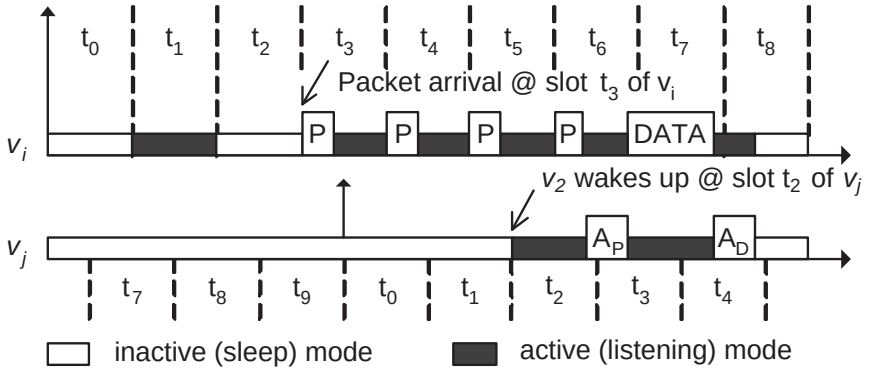


Figure 5.5: Probing mechanism in the asynchronous MAC model.

it wakes up and starts to broadcast small probes P at each subsequent slot to indicate to any of its awake neighbors that it has data to forward to the fusion center v_0. When a potential forwarder v_j in the forwarding set wakes up, it stays awake for a short period of time (of at least one slot time τ) to receive any incoming probes. When v_j receives a probe from its upstream node v_i, it responds by transmitting a probe acknowledgement A_P to v_i, after a small delay. This delay is randomized to minimize collisions when multiple forwarders of v_i are awake at the same time and receive the probe packet. When v_i receives A_P from at least one forwarder, it forwards the data packet to it. Upon receipt of the data packet, v_j sends a corresponding data acknowledgement A_D. v_i resumes its wakeup schedule after receiving A_D from v_j, while the latter has to continue forwarding the packet to its downstream nodes towards the fusion center.

The event detection model in Chapter 4.3 is adopted, whereby hypothesis H_1 denotes presence of the PoI in the sensor network and H_0 denotes the corresponding absence of the PoI. Sensor observations are i.i.d. at each sensor and across sensors when conditioned upon the hypothesis H_i, $i \in \{0, 1\}$.

At periodic sensing intervals, each node v_i observes an independent signal y_i

given by:

$$y_i = \begin{cases} w_i & \text{if } H_0 \text{ (PoI is absent);} \\ f(r_i) + w_i & \text{if } H_1 \text{ (PoI is present),} \end{cases} \tag{5.1}$$

where $w_i \sim \mathcal{N}(\mu_w, \sigma_w^2)$ is the white Gaussian noise seen by v_i that follows a normal distribution with mean μ_w and standard deviation σ_w; r_i is the the distance between v_i and the PoI; and f is a function that monotonically decreases with increasing r_i. Recall that one such function is the exponential sensing model [143] [144] defined by:

$$f(r_i) = \begin{cases} s_0 & \text{if } r_i \leq r_a; \\ s_0 \cdot e^{-\delta(r_i - r_a)} & \text{otherwise,} \end{cases} \tag{5.2}$$

where r_a is the sensing range below which the signal generated by the PoI starts to undergo attenuation; s_0 is the signal strength of the PoI when measured within a distance of r_a from the PoI; and δ is the sensing capacity decay (reflecting the attenuation of the signal generated by the PoI).

For each observed signal y_i, v_i makes a **per-sample binary decision** $b_i \in \{0, 1\}$ such that:

$$b_i = \begin{cases} 0 & \text{if } y_i < T; \\ 1 & \text{otherwise,} \end{cases} \tag{5.3}$$

where T is the per-sample threshold. The set of activated nodes that detect the PoI ($b_i = 1$) is denoted as $V_a \subset V$.

The optimal fusion rule for fusion center v_0 using data from *all* the activated nodes when PoI is present is the Likelihood Ratio Test (LRT) [141] [142]:

$$\Lambda(B) = \frac{P(b_1, b_2, ..., b_{|V_a|}|H_1)}{P(b_1, b_2, ..., b_{|V_a|}|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{1-p}{p}, \tag{5.4}$$

where $0 < p < 1$ is the *a priori* probability $P(H_1)$; and $B = \bigcup_{v_i \in V_a} b_i$ is the set of per-sample binary decisions from all the activated nodes in V_a. The targeted detection and false alarm probabilities are denoted as P_d and P_f respectively. Based on Sequential Probability Ratio Test (SPRT) [140], data acquisition for reliable PoI detection can terminate **as soon as** the cumulative log-likelihood ratio S_0 at v_0 satisfies a particular IQ threshold $I_T = \log \frac{P_d}{P_f}$, such that:

$$S_0 = \log \Lambda(B_k) = \log \prod_{b_i \in B_k} \Lambda(b_i) \geq I_T, \tag{5.5}$$

where $B_k \subseteq B$ is the **earliest ordered subsequence** of data collected from the activated nodes. The information quality q_i provided by each activated node v_i can then be considered as its log-likelihood ratio, such that:

$$q_i = \log \Lambda(b_i) = \log \frac{P(b_i|H_1)}{P(b_i|H_0)}. \tag{5.6}$$

5.3.2 PoI Detection Delay with IQ-Awareness

Let Δ_i be the **aggregation latency** that a node v_i waits for after the first data arrival, to allow data (if any) from its upstream nodes to be combined together before forwarding the aggregated data to fusion center v_0.

With the use of SPRT and IQ-awareness, only a subset of the activated nodes is required to detect the PoI with sufficient reliability and accuracy. We let n_a be the number of such subsets of activated nodes whose aggregated IQ meets the IQ threshold I_T, where n_a is bounded by $2^{|V_a|}$. Each of these subsets $V_i^a \subseteq V_a$ can then be uniquely labeled as $V_1^a, V_2^a, ..., V_{n_a}^a$, such that the aggregated information quality from all the nodes in each subset is greater than the IQ threshold, i.e. $\sum_{v_j \in V_i^a} q_j \geq I_T \; \forall 1 \leq i \leq n_a$. Then, the **minimum PoI detection delay** D_p is the time required to get data from all the activated nodes in *any* of the subsets $V_i^a \subseteq V_a$. Obviously, D_p depends on the values of Δ_i.

There may be multiple routes to collect data from all the nodes in each

subset V_i^a; we denote the set of all possible routes for V_i^a as $R_i = \bigcup R_i^j$. The corresponding PoI detection delay incurred by each route R_i^j is denoted $D[R_i^j]$. Then, the minimum PoI detection delay incurred using the subset of activated nodes V_i^a is given by:

$$D[V_i^a] = \min_{R_i^j \in R_i} D[R_i^j]. \tag{5.7}$$

The minimum PoI detection delay D_p for the entire network is given by the minimum delay required for *any* of the subsets $V_i^a \subseteq V_a$:

$$D_p = \min_{V_i^a \subseteq V_a} D[V_i^a]. \tag{5.8}$$

5.3.3 Problem Formulation

Suppose we are given a network $G = \{V, E\}$; duty cycle of each node $v_i \in V$; set of activated nodes $V_a \subset V$; IQ contribution q_a of each activated node $v_a \in V_a$; and IQ threshold I_T required for reliable and accurate PoI detection. Each link e_{ij} between two arbitrary nodes $v_i, v_j \in V$ is associated with a per-hop delay that is dependent on the aggregation latency Δ_i of the transmitter v_i and duty cycle of the receiver v_j. The key objectives of our work are to: (i) dynamically and distributively construct an IQ-aware aggregation tree in real-time; and (ii) assign a corresponding aggregation latency Δ_i to each node v_i in the aggregation tree, such that the PoI can be detected with an aggregated IQ of at least I_T using the least amount of energy and with the minimum PoI detection delay D_p.

Formally, this is equivalent to finding a minimum delay *subset-τ* Steiner Tree (SST)[2], which is a Steiner Tree $G_\tau = \{V_\tau, E_\tau\} \subseteq G$ that spans the fusion center v_0 and all nodes in $V_\tau \subseteq V_a$, such that:

- aggregated IQ collected from the activated nodes in V_τ exceeds I_T; and
- using G_τ as the aggregation tree gives the minimum PoI detection delay among all possible *subset-τ* Steiner trees that meet the IQ constraint.

[2]The SST problem is defined earlier in Chapter 4.3.

We denote the delay[3] between v_0 and an arbitrary node $v_i \in V_\tau$ as $D[v_i]$; the corresponding PoI detection delay using V_τ is given by $D[V_\tau] = \max_{v_i \in V_\tau} D[v_i]$. Thus, we want to find the Steiner Tree G_τ that gives:

$$\min_{V_\tau \subseteq V_a} \; D[V_\tau]; \tag{5.9}$$

subject to:

$$\sum_{v_i \in V_\tau} q_i \geq I_T; \tag{5.10}$$

$$\Delta_i \geq 0 \quad \forall v_i \in V_\tau; \tag{5.11}$$

and

$$\mathcal{C}(e_{ij}) = 1 \quad \forall e_{ij} \in E_\tau, \tag{5.12}$$

where $\mathcal{C}(e_{ij})$ denotes the number of times that edge e_{ij} has been traversed. The first constraint (Equation 5.10) specifies the IQ requirement. The second constraint (Equation 5.11) specifies the feasible space for aggregation latencies. The third constraint (Equation 5.12) expresses the goal of minimizing transmissions and maximizing aggregation efficiency in order to minimize energy consumption.

Note that $D[V_\tau]$ is the minimum PoI detection delay incurred using all possible aggregation paths that span all the activated nodes in V_τ. The key parameters to be determined are the set of Δ_is.

However, finding such a minimum delay *subset-τ* Steiner Tree for an event driven sensor network in real-time is inherently difficult and intractably NP-hard, for reasons which we detail as follows:

1. Computing the optimal aggregation structure that provides the global minimum PoI detection delay D_P requires each node to acquire knowledge of: (i) global network topology; (ii) global duty cycling wakeup schedule; and (iii) IQ contributions of each activated node.

[3]Note that if all nodes use the same duty cycle, then this delay is proportional to the path length.

2. However, the PoI can occur at any point in time during network lifetime, and at any location in the monitored terrain. Hence, the set of activated nodes $V_a \subset V$ as well as location and individual IQ contribution q_a of each activated node $v_a \in V_a$ are not known *a priori*.

3. Acquiring this information in real-time incurs excessive overhead, which can have detrimental impact on resource-constrained sensor networks.

4. The number of sets of activated nodes $V_i^a \subseteq V_a$ whose aggregated information quality satisfies the IQ threshold I_T is in the order of $n_a = O(2^{|V_a|})$. For each of these n_a subsets, there exists multiple possible aggregation paths to the fusion center. Subsequently, it is non-trivial to construct the optimal minimum delay *subset-*τ Steiner Tree that provides both minimum PoI detection delay and minimum energy consumption, in a distributed and efficient manner in real-time.

In the next section, we detail **IQDEA** - an *I*nformation *Q*uality aware *D*elay *E*fficient Aggregation scheme that provides an efficient and distributed methodology to construct IQ-aware aggregation trees that incur small PoI detection delays and low transmission costs in real-time.

5.4 Methodology

In an event driven sensor network, data arrival takes place due to: (i) data generation resulting from PoI detection (i.e. $v_i \in V_a$); or (ii) data reception from one or more upstream node(s). Upon data arrival, v_i must decide on:

aggregation latency Δ_i: Computed upon the *first data arrival*, this is the length of time to wait for potential data to arrive from upstream node(s), so that data may be aggregated before being forwarded to the fusion center via the downstream node.

forwarding decision f_i: Computed when v_i is ready to forward data towards the fusion center, this is the downstream node that v_i selects to transmit its aggregated data to.

This section describes how IQDEA selects the aggregation latency and forwarder of each node to achieve good energy-delay tradeoffs, while satisfying information quality constraints at fusion center v_0.

5.4.1 Aggregation Latency

Computing Aggregation Latency

The aggregation latency Δ_i is determined with respect to the first data arrival at v_i. If another data packet arrives at v_i when its data buffer is not empty, aggregation takes place and Δ_i need **not** be recomputed.

We make the following observations which are useful in the choice of aggregation latency:

O_1**:** To prevent routing loops and minimize path length, data typically flows unidirectionally from a node with larger routing metric towards a node with lower routing metric.

O_2**:** As the PoI location is not known *a priori*, existing structured aggregation schemes compute the aggregation latency of each node based on the worst case scenario (that data is required from all nodes in the network for PoI detection and the maximum per-hop delay is incurred at each hop), leading to substantially longer PoI detection delays.

O_3**:** Aggregation schemes without IQ-awareness assume that all activated nodes in the network are required for PoI detection, thereby incurring higher energy expenditure and possibly longer PoI detection delays than necessary.

O_4**:** Per-node IQ of a node generally decreases with increasing distance from

PoI[4].

Based on each of these observations, we draw the corresponding inferences:

I_1: Aggregation opportunities can be maximized (hence reducing energy consumption) if aggregation latency decreases linearly with the routing metric of the node. This allows data from node v_i with larger routing metric to be aggregated at node v_j with smaller routing metric, before v_j forwards data to v_0.

I_2: To reduce PoI detection delay, the aggregation latency of each node should not be computed solely based on its location and network diameter.

I_3: The optimal aggregation latency of each node is the minimum delay required to obtain data from its activated (direct and indirect) upstream nodes from a subset of the data collection tree (i.e. minimum delay subset-τ Steiner Tree), and *not* all the activated nodes in the network. However, nodes that are in this tree are not known *a priori*, and obtaining such knowledge in real-time incurs excessive communication overheads.

I_4: Aggregation latency cannot be computed based on IQ alone, as per-node IQ generally decreases in a concentric circles centered around the PoI, instead of a monotonic path towards v_0.

We first consider two aggregation latency functions, viz. linear-based[5] $L(h) \rightarrow \mathbb{R}$ and exponential-based $E(h) \rightarrow \mathbb{R}$, as illustrated in Figure 5.6. With the use of $L(h)$, each node along the aggregation tree waits for the maximum amount of time to maximize aggregation opportunities. The difference between $L(h)$ and $E(h)$, i.e. $L(h) - E(h)$, provides an approximation of the reduction in aggregation latency (and corresponding PoI detection delay) when $E(h)$ is used

[4]Although this may not be true for all sensing models, typical physical sensing modalities such as noise, pressure and visual light follow the exponential sensing model which exhibit such characteristics.

[5]The linear-based function $L(h)$ is commonly used in many data aggregation schemes.

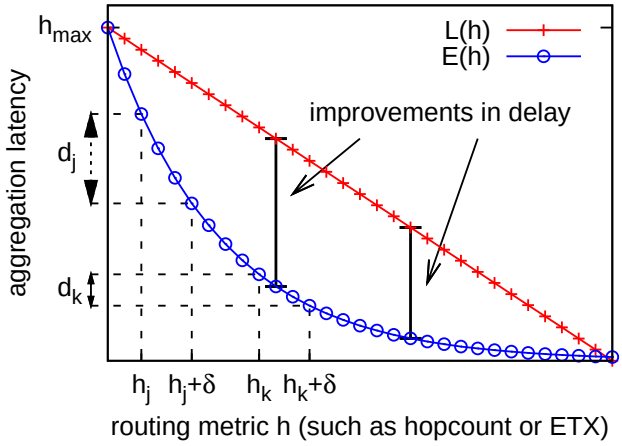


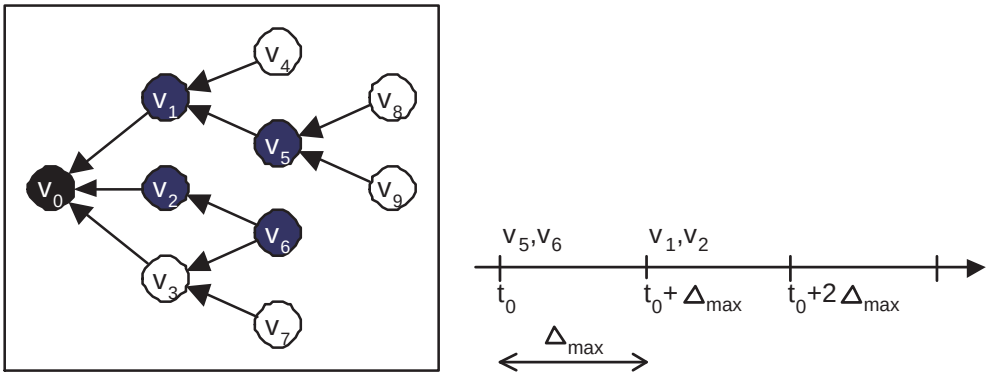
Figure 5.6: Effect of Using Different Aggregation Latency Functions.

in place of $L(h)$. Hence, $L(h) - E(h)$ should be maximized to achieve greater reductions in aggregation latency and PoI detection delay, and a function of the form $E(h) = e^{-h}$ would seem appropriate.

However, data aggregation can take place only if there is a minimum difference in aggregation latency between two nodes whose routing metrics differ by some amount - say δ. For an arbitrary value of h in Figure 5.6, the difference in aggregation latency $L(h) - L(h + \delta)$ is a constant that is chosen to ensure that maximal aggregation opportunities take place. With the use of exponential-based aggregation function $E(h)$, the aggregation latency difference $E(h) - E(h + \delta)$ varies according to the value of h that is used. With large values of h, $E(h) - E(h + \delta)$ becomes very small and may not be sufficiently long for data aggregation to take place successfully. For example, $d_j > d_k$ in Figure 5.6. Thus, using an exponential-based $E(h)$ as an aggregation function may not achieve both energy and delay efficiency in data aggregation. This is particularly true when the PoI is further away from the fusion center.

We now describe the intuition behind the aggregation latency used in IQDEA, which is based on observations $O_1 - O_4$ and inferences $I_1 - I_4$, from which we note that a good heuristic (Heuristic H_{good}[6]) to approximate near-optimal energy-

[6] H_{good} is optimal if data from all activated nodes are required at the fusion center.



(a) Aggregation paths with set of activated nodes $V_a = \{v_1, v_2, v_5, v_6\}$.

(b) Timeline of scheduled transmissions using Heuristic H_{good}.

Figure 5.7: Illustration of the delays incurred by a structured aggregation tree.

delay tradeoff is to: (i) initiate data aggregation from the *activated* node with largest routing metric $h^a_{\max}$; and (ii) stagger the aggregation latency of each node according to its routing metric, such that the (furthest) activated node with metric $h^a_{\max}$ has an aggregation latency of 0, and each intermediate node v_j (with metric $h_j < h^a_{\max}$) involved in the forwarding process has an aggregation latency of $(h^a_{\max} - h_j) \cdot \Delta_{\max}$.

Based on Heuristic H_{good}, activated nodes v_5 and v_6 which have the largest routing metric in Figure 5.7(a) should transmit at the start time of the cycle t_0. As illustrated in Figure 5.7(b), v_1 and v_2 (which are closer to fusion center v_0) should transmit at time $t_0 + \Delta_{\max}$ after aggregating data from v_5 and v_6 respectively. Thus, v_0 is able to receive data from the activated nodes after a delay of $2\Delta_{\max}$ and a total of 4 transmissions.

We now consider H_{good} in a generalized network setting. Suppose the following information is known:

$h^a_{\max}$**:** maximum routing metric of any activated node; and

h^a_{Δ}**:** maximum difference in routing metric between any two activated nodes in the network.

As illustrated in Figure 5.8, an activated node v_i with routing metric $h_i =$

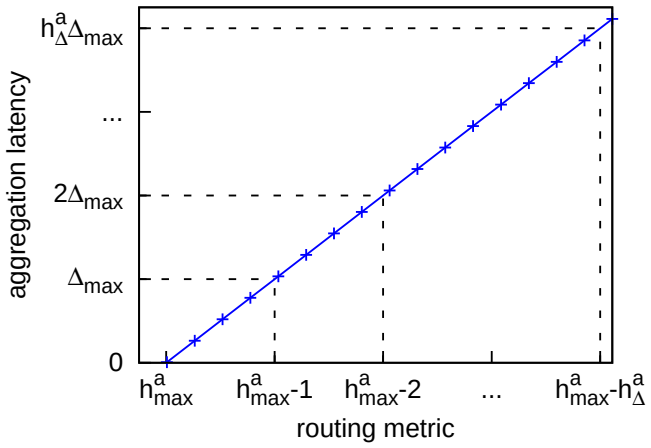


Figure 5.8: Aggregation latency using Heuristic H_{good}.

$h^a_{\max} - j$ should then have an aggregation latency of $\Delta_i = (h^a_\Delta - j) \cdot \Delta_{\max}$, where $\Delta_{\max}$ is the maximum per-hop delay.

A key property that can be observed from the aggregation latency derived in H_{good} is that it is bounded by $h^a_\Delta \cdot \Delta_{\max}$. This can be justified as follows: Since the maximum difference in routing metric between any two activated nodes is h^a_Δ, it is not necessary for any node to have an aggregation latency beyond this threshold.

However, we note that H_{good} is:

- not delay-energy optimal as it acquires data from all activated nodes; and
- not practical as the location(s) of node(s) with the largest routing metric is not known *a priori*.

The following describes how IQDEA applies the bounded aggregation latency in H_{good} in a realistic network setting whereby activated nodes in an event driven sensor network are not known in advance.

We assume that each node v_i has knowledge of the IQ threshold I_T; per-sample threshold T; monotonic sensing model $f(r_i)$ which affects signal y_i received at v_i; routing metric h_i of v_i; and maximum value of the routing metric $h_{\max}$ in the network.

When the PoI occurs, an activated node $v_i \in V_a$ can compute its IQ contribution $q_i = \log \Lambda(b_i)$, according to Equation 5.6. It then estimates the maximum distance $E[r_{\max}]$ of any activated node from the PoI as follows:

$$E[r_{\max}] = f^{-1}(T - w_i), \tag{5.13}$$

where f^{-1} is the inverse function of the monotonic sensing model; and $w_i \sim \mathcal{N}(\mu_w, \sigma_w^2)$ is Gaussian noise with mean μ_w and standard deviation σ_w.

We denote the minimum and maximum values of the routing metric of an *activated* node as $h^a_{\min}$ and $h^a_{\max}$ respectively. The next step is to estimate the maximum difference in routing metric between any two activated nodes based on the sensing model $f(r_i)$ in Equation 5.2 and estimated maximum distance $E[r_{\max}]$ of any activated node from the PoI in Equation 5.13. In general, we require a distance-based routing metric, such as hopcount, ETX or physical distance (geographical routing). In the case of ETX, good links with high packet reception ratios (PRRs) are preferred. If the selected links have high PRRs of at least 0.9, ETX can be considered to be a close approximation of hopcount. For simplicity, we assume that the routing metric used by IQDEA is hopcount. Taking average transmission range to be r, the expected normalized progress per hop $\hat{p}$ can be estimated as [147]:

$$\hat{p} = 1 + e^{-N} - \int_{-1}^{1} e^{-\frac{N}{\pi}(\cos^{-1} t - \sqrt{1-t^2})} dt, \tag{5.14}$$

where N is average number of nodes in a transmission region. Figure 5.9 illustrates how the expected normalized progress per hop varies with N.

The expected maximum difference in hopcount h^a_Δ between any two activated nodes can then be approximated by:

$$h^a_\Delta = E[h^a_{\max} - h^a_{\min}] \approx \lceil \frac{2 \cdot E[r_{\max}]}{\hat{p} \cdot r} \rceil, \tag{5.15}$$

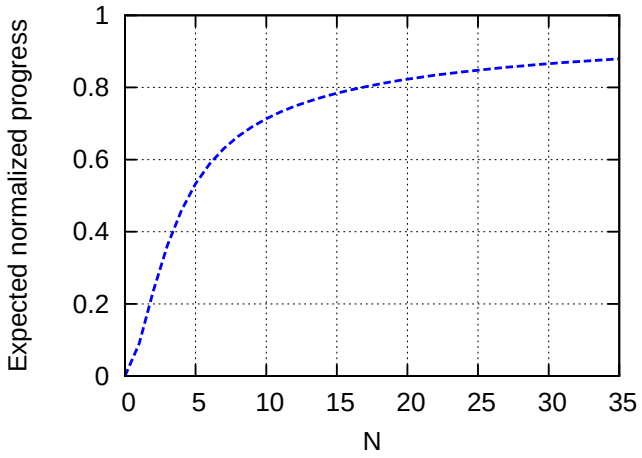


Figure 5.9: Expected normalized progress per hop as a function of number of nodes in the transmission range N.

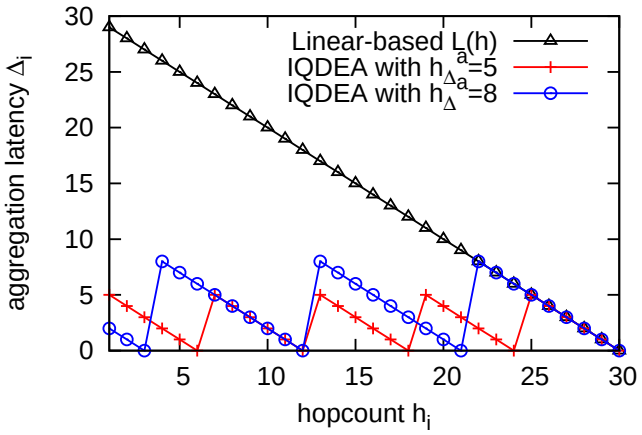


Figure 5.10: Aggregation latency as a function of routing metric h (with maximum per-hop delay $\Delta_{\max} = 1$).

where r is the transmission range.

The aggregation latency Δ_i of each **activated** node $v_i \in V_a$ (normalized to the time of PoI occurrence) is then defined as:

$$\Delta_i = [(h_{\max} - h_i) \mod (h^a_\Delta + 1)] \cdot \Delta_{\max}. \tag{5.16}$$

Recall that $\Delta_{\max}$ is the expected maximum per-hop delay incurred by any node in the network.

Figure 5.10 illustrates how the aggregation latency of each activated node varies according to its routing metric h for different values of h^a_Δ. As compared to the linear-based function $L(h)$ shown in Figure 5.6, we can observe that:

- difference in aggregation latency incurred by $L(h)$ and IQDEA is substantial, particularly for small values of h - thus demonstrating that IQDEA can significantly reduce the latencies incurred by $L(h)$;
- difference in aggregation latency between two nodes with routing metrics h and $h+1$ is a constant[7] - thereby enabling sufficient time for data aggregation and hence improving energy efficiency in IQDEA; and
- good latency performance is achieved across all PoI locations.

In order to minimize collisions resulting from concurrent transmissions by nodes with the same routing metric and to give higher priority to nodes (of the same metric) with higher information quality, the aggregation latency of v_i is incremented by a factor of its IQ q_i, such that:

$$\Delta_i = [(h_{\max} - h_i) \mod (h^a_\Delta + 1) + k \cdot \frac{q_{\max} - q_i}{q_{\max} - q_{\min}}] \cdot \Delta_{\max}, \tag{5.17}$$

where $0 < k < 1$ is the weight allocated to IQ prioritization; $q_{\min}$ is the minimum IQ that is obtainable in the activated region; and $q_{\max}$ is the maximum IQ that is obtainable in the activated region.

The values $q_{\min}$ and $q_{\max}$ can be computed as follows:

$$q_{\min} = \log(\frac{Q(\frac{T - f(E[r_{\max}])}{\sigma_w})}{Q(\frac{T}{\sigma_w})}); \tag{5.18}$$

and

$$q_{\max} = \log(\frac{Q(\frac{T - f(0)}{\sigma_w})}{Q(\frac{T}{\sigma_w})}), \tag{5.19}$$

[7] $h \in \mathbb{Z}^+$ and $h \neq m \cdot (h^a_\Delta + 1)$ where $m \in \mathbb{Z}^+$.

where $Q(x)$ is the Gaussian Q-function of a standard normal distribution. Note that $Q(\frac{T_i}{\sigma_w})$ is equivalent to the per-sample probability of false alarm of v_i.

We denote the aggregated IQ at v_i as Q_i. Then, the aggregation latency Δ_i to be computed at an arbitrary node v_i with hopcount h_i in the network can be summarized as follows:

$$\Delta_i = \begin{cases} 0 & \text{if } Q_i \geq I_T \text{ or } q_i = 0; \\ [(h_{\max} - h_i) \mod (h^a_\Delta + 1) + k \cdot \frac{q_{\max} - q_i}{q_{\max} - q_{\min}}] \cdot \Delta_{\max} & \text{otherwise.} \end{cases} \quad (5.20)$$

Based on Equation 5.20, aggregation latency is not incurred when: (i) sufficient data has been aggregated at v_i; or (ii) v_i is not an activated node. This allows data to be forwarded towards fusion center v_0 quickly when the PoI can be detected reliably based on existing aggregated data (i.e. $Q_i \geq I_T$), thus reducing PoI detection delay at v_0. On the other hand, when data is being forwarded by a non-activated node (i.e. $q_i = 0$), it is likely that the aggregated data is no longer within the activated region. Consequently, it is unlikely for aggregation opportunities and IQ contributions to arrive from upstream nodes within the next aggregation cycle, and existing aggregated data should be forwarded to v_0 as soon as possible to minimize PoI detection delay.

The effectiveness of the given aggregation scheme requires knowledge of the maximum difference in routing metric (or hopcount) h^a_Δ between any two activated nodes, which cannot be obtained *a priori* but has to be estimated. We show in our performance evaluation that the algorithm is robust and good performance can be obtained even if there exists some errors in the estimation.

Delay Bounds

In this section, we present performance bounds for PoI delay detection. We can view the monitored terrain as a series of concentric circles centered at v_0 (Figure 5.11), each of radius $(h^a_\Delta + 1)$ smaller than its adjacent larger (outer) circle. The

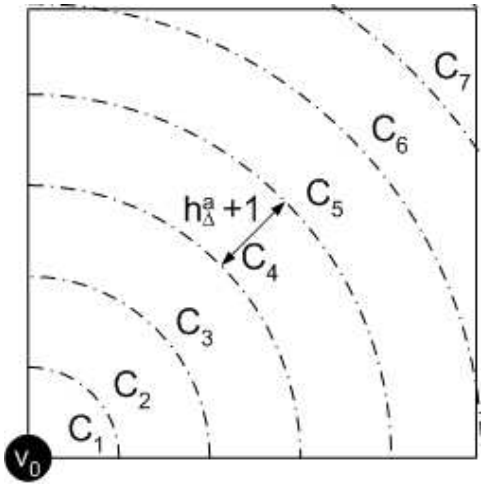


Figure 5.11: Network is divided into concentric circles centered at the fusion center v_0. The radius of each circle differs from its adjacent circle by h^a_Δ.

number of such concentric circles is given by $n_r = \lceil \frac{h_{\max}}{h^a_\Delta + 1} \rceil$; each concentric circle can then be uniquely labeled as C_1, C_2, ..., C_{n_r}. The hopcounts of nodes within each circle C_i $(1 \leq i \leq n_r)$ is lower bounded by:

$$h(C_i^L) = \max(0, h_{\max} - (n_r - i + 1)(h^a_\Delta + 1) + 1); \tag{5.21}$$

and upper bounded by:

$$h(C_i^U) = h_{\max} - (n_r - i)(h^a_\Delta + 1). \tag{5.22}$$

We now consider an arbitrary concentric circle C_i. Suppose the PoI occurs within the concentric circle C_i, such that the hopcounts of the activated nodes span between $h(C_i^L)$ and $h(C_i^U)$. The aggregation latency of each node within C_i is then staggered according to its hopcount - such that node v_j with hopcount $h_j = h(C_i^U)$ has the least aggregation latency $\Delta_j = 0$ and node v_k with hopcount $h_k = h(C_i^L)$ has the largest aggregation latency $\Delta_k = h^a_{\max} \cdot \Delta_{\max}$. This is equivalent to the aggregation latency function as defined in Equation 5.16. It is trivial to see that when all the activated nodes fall within a single concentric circle, this

aggregation latency model is similar to Heuristic H_{good} that approximates the near-optimal energy-delay tradeoff.

In realistic scenarios, the PoI can occur at any location in the network; thus, the activated nodes in the network may span across more than a single concentric circle. As the expected maximum difference in hopcount between any two activated nodes is h^a_Δ and the radii of adjacent concentric circles differ by $h^a_\Delta + 1$, the maximum number of concentric circles spanned by the activated nodes is at most 2 according to Lemma 4. The corresponding maximum penalty in terms of additional aggregation latency incurred as compared to H_{good} is therefore bounded by $h^a_\Delta \cdot \Delta_{\max}$, based on Lemma 5.

Lemma 4. *Let the expected maximum difference in hopcount between any two activated nodes be h^a_Δ. The radii of adjacent concentric circles differ by $h^a_\Delta + 1$. Then, regardless of the location of the PoI, the maximum number of concentric circles spanned by activated nodes is 2.*

Proof. We prove Lemma 4 by contradiction. Suppose the number of concentric circles spanned by activated nodes is more than 2. Then, the maximum difference in hopcounts between any two activated nodes is at least $h^a_\Delta + 1 + 1 = h^a_\Delta + 2$, which is greater than the expected maximum difference in hopcounts h^a_Δ. □

Lemma 5. *The maximum penalty in terms of additional aggregation latency incurred as compared to H_{good} is bounded by $h^a_\Delta \cdot \Delta_{\max}$.*

Proof. A penalty in aggregation latency is incurred when the minimum hopcount $h^a_{\min}$ of an activated node does not fall on the lower bound of the concentric circle in which it lies in, i.e., $(h_{\max} - h^a_{\min}) \mod (h^a_\Delta + 1) < h^a_\Delta$. This causes the activated nodes to span two concentric circles. The additional aggregation latency incurred is the time taken to transmit the aggregated data from the node with lowest hopcount in the outer concentric circle to the activated node with the minimum hopcount (in the inner concentric circle), which is bounded by $h^a_\Delta \cdot \Delta_{\max}$ as the radii difference of each circle is $h^a_\Delta + 1$. □

Based on observations $O_3 - O_4$ and inferences $I_3 - I_4$, (i) utilization of information quality awareness in data aggregation schemes eliminates the need for data to be collected from all the activated nodes in the network; and (ii) priority in transmissions should therefore be given to nodes which have higher IQ. Data from activated nodes are suppressed when sufficient data has been collected for reliable and accurate PoI detection at fusion center. As such, even if the activated nodes in the network span across two concentric circles, penalties in aggregation latency may not be incurred if sufficient data for PoI detection has been collected within the inner circle.

Baseline Comparison

We now present a simple comparison of the PoI detection delays incurred by the three aggregation schemes - structured aggregation, structureless (opportunistic) aggregation and IQDEA - based on their corresponding aggregation latency functions. The PoI detection delay comprises of: (i) aggregation latency from the furthest activated node (with hopcount $h^a_{\max}$) to the nearest activated node (with hopcount $h^a_{\min}$); and (ii) forwarding latency from the nearest activated node to the fusion center.

In structured aggregation schemes (denoted as AggTree), the maximum aggregation latency is incurred by the activated node that is nearest to the fusion center. The PoI detection latency incurred is given by:

$$\begin{aligned} D_p(AggTree) &= (h_{\max} - h^a_{\min}) \cdot \Delta_{\max} + h^a_{\min} \cdot \Delta_{\max} \\ &= h_{\max} \cdot \Delta_{\max}. \end{aligned} \tag{5.23}$$

Structureless (opportunistic) aggregation schemes do not incur any aggregation latencies. The PoI detection delay incurred by these schemes is bounded by

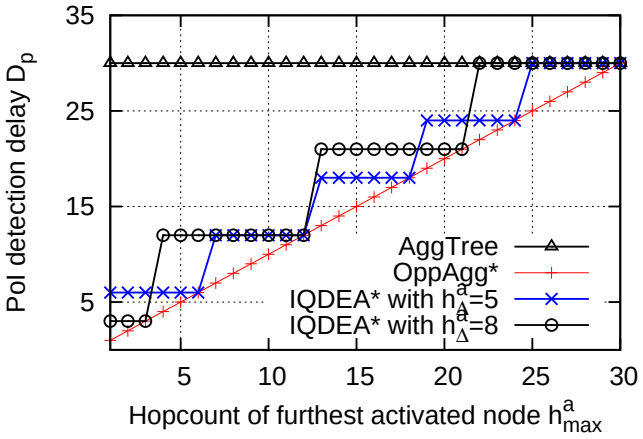


Figure 5.12: PoI detection delay D_p under different aggregation schemes (with network diameter $h_{\max} = 30$ and maximum per-hop delay $\Delta_{\max} = 1$).

the forwarding latency required to collect data from all the activated nodes:

$$\begin{aligned} D_p(OppAgg) &\leq 0 + \max_{v_i \in V_a}(h_i \cdot \Delta_{\max}) \\ &\leq h^a_{\max} \cdot \Delta_{\max}. \end{aligned} \tag{5.24}$$

Finally in IQDEA, the PoI detection delay is bounded by the total time taken to aggregate data from all the activated nodes in the network. Note that the actual detection delay is typically less due to IQ-awareness in IQDEA, whereby the PoI can be detected as soon as sufficient data been acquired.

$$D_p(IQDEA) \leq [(h_{\max} - h^a_{\max}) \mod (h^a_\Delta + 1) + h^a_{\max}] \cdot \Delta_{\max}. \tag{5.25}$$

Figure 5.12 compares the PoI detection delay D_p incurred by various aggregation schemes, when the hopcount of the furthest activated node $h^a_{\max}$ increases. Note that OppAgg* and IQDEA* indicate the upper bounds of the detection delays using these two aggregation schemes.

5.4.2 Forwarder Selection

When aggregation latency $\Delta_i = 0$ at v_i, it is ready to forward data to the fusion center v_0 through one of its downstream nodes. The forwarding decision f_i determines the downstream node that v_i will transmit to, and is based on: (i) *expected* end-to-end delay from v_i to v_0 via each downstream node; and (ii) aggregation opportunities at each downstream node.

Recall that the set of one-hop neighbors of v_i is denoted as N_i, and the corresponding set of potential forwarders of v_i is denoted as $F_i \subseteq N_i$, where $h_j < h_i \ \forall v_j \in F_i$. The aggregated IQ at $v_j \in F_i$ is denoted as $Q_j \geq 0$. The expected minimum delay of v_i to v_0 via $v_j \in F_i$ is denoted as $E[d_i(j)]$ and the corresponding expected minimum delay of v_i to v_0 is denoted as $E[d_i]$.

We now describe how $E[d_i(j)]$ and $E[d_i]$ are computed. During network initialization, $E[d_i(j)] = \infty$ and $E[d_i] = \infty \ \forall v_i \in V, v_j \in F_i$. The fusion center v_0 then updates $E[d_0] = 0$ and broadcasts this value to its neighbors.

Upon receiving an update $E[d_j]$ from $v_j \in F_i$, v_i updates $E[d_i(j)]$ and $E[d_i]$ as follows:

$$E[d_i(j)] = E[d_j] + E[s_i(j)]; \tag{5.26}$$

and:

$$E[d_i] = \min_{v_j \in F_i} E[d_i(j)], \tag{5.27}$$

where $E[s_i(j)]$ is the expected sleep latency incurred by v_i when transmitting to v_j. Here, it should be noted that no sleep latency is incurred when: (i) the network is always-on (without duty cycling), i.e. $E[s_i(j)] = 0 \ \forall v_i, v_j \in V$; or (ii) transmitting to the fusion center v_0 (which is assumed to be always-on), i.e. $E[s_i(0)] = 0 \ \forall v_i \in V$. In addition, the **instantaneous expected sleep latency** $E[s'_i(j)] = 0$ if the receiving node v_j is currently awake. For an asynchronous duty cycled MAC protocol such as that used in IQDEA, the expected sleep latency experienced by v_i when transmitting to v_j is dependent on the duty

Algorithm 5 Computing minimum expected delay $E[d_i]$.

Input: $G = \{V, E\}$; $\alpha_i \ \forall v_i \in V$
Variable: expected delay from v_i to v_0 through v_j $E[d_i(j)] = \infty$; expected delay from v_i to v_0 $E[d_i] = \infty \ \forall v_i \in V, v_j \in F_i$
Output: $E[d_i(j)]$, $E[d_i]$
v_0 updates and broadcasts $E[d_0] = 0$
while v_i receives $E[d_j]$ from $v_j \in F_i$ **do**
    $E[d_i(j)] = E[d_j] + \frac{\tau(n_c - \alpha_j)}{\alpha_j}$
    $E[d_i] = \min_{v_j \in F_i} E[d_i(j)]$
    if $E[d_i]$ is updated **then**
        broadcast $E[d_i]$
    end if
end while

cycling wakeup schedule and follows the geometric distribution such that:

$$E[s_i(j)] = \frac{\tau(n_c - \alpha_j)}{\alpha_j}, \tag{5.28}$$

where τ is the slot length; n_c is the number of slots in each cycle; and $\frac{\alpha_j}{n_c}$ is the active probability of the receiving node v_j as defined earlier in Chapter 5.3.1.

The value of $E[d_i]$ is broadcasted by v_i to its neighbors if it is updated. Eventually, $E[d_i] \ \forall v_i \in V$ converges after $h_{\max}$ iterations, where $h_{\max}$ is the network diameter. Algorithm 5 summarizes the procedure used by each node to compute its minimum expected delay $E[d_i]$.

Observe that $E[d_i]$ and $E[d_i(j)]$ remain the same if duty cycles of nodes along the paths remain the same. However, the **instantaneous value** of $d_i(j)$ and hence d_i changes whenever: (i) a downstream node $v_j \in F_i$ wakes up; and/or (ii) an intermediate forwarding node along any of the possible paths from $v_j \in F_i$ to v_0 wakes up. However, it is both costly and impractical to update $E[d_i(j)]$ and $E[d_i]$ whenever any of these changes occur, especially when the network is dense and/or network diameter is large. To reduce overheads, the **instantaneous expected delay** $E[d'_i(j)]$ from v_i to v_0 through v_j is updated to be $E[d'_i(j)] = E[d_j]$ only when: (i) v_i is ready to forward data to v_0; and (ii)

$v_j \in F_i$ wakes up[8].

Node v_i then makes the forwarding decision $f_i = v_j \in F_i$ if any one of the following conditions is satisfied:

C_1**:** There exists aggregated data at v_j, and it is the only potential forwarding node that is currently awake.

C_2**:** Among all the potential forwarding nodes that are currently awake, the aggregated data at v_j has the highest aggregated information quality.

C_3**:** There is no aggregated data at v_j. However, v_j is currently awake and has the least instantaneous expected delay towards the fusion center as compared to all the other forwarding nodes.

These conditions can be summarized in mathematical notations as follows:

C_1: $Q_j > 0$, $\Delta'^s_{ij} = 0$ and $\Delta'^s_{ik} > 0$ $\forall v_j, v_k \in F_i$.

C_2: $Q_j > Q_k$, $\Delta'^s_{ij} = \Delta'^s_{ik} = 0$ $\forall v_j, v_k \in F_i$.

C_3: $Q_j = 0$, $\Delta'^s_{ij} = 0$, $E[d'_i(j)] \leq E[d'_i(k)]$ $\forall v_j, v_k \in F_i$.

If v_j is not selected as a forwarding node in the current time slot and there are no other awake forwarders, then v_i will continue waiting for another forwarder to be awake in the subsequent time slots.

5.5 Performance Evaluation

We consider the routing metric of each node to be its hopcount from fusion center v_0 and evaluate the performance of the following data aggregation schemes on GloMoSim [121]:

1. AggTree: Structured aggregation tree whereby each node v_i waits for aggregation latency Δ_i that is inversely proportional to its hopcount. This

[8]The **instantaneous sleep latency** $\Delta'^s_{ij} = 0$ as there is no sleep latency incurred by v_i when v_j is awake.

scheme is expected to provide highest energy efficiency in terms of transmission and longest PoI detection delay.

2. OppAgg: Structureless aggregation scheme whereby each node v_i forwards data towards fusion center immediately upon data arrival (i.e. $\Delta_i = 0$). The scheme is expected to provide earliest PoI detection and most number of transmissions.

3. IQDEA: Aggregation scheme that provides fast PoI detection and uses IQ-awareness to terminate data acquisition when sufficient data has been acquired at fusion center. Aggregation delay at each node v_i is computed based on its hopcount h_i, estimated maximum hopcount difference h^a_Δ between activated nodes and network diameter $h_{\max}$. This scheme is expected to provide good energy-delay tradeoffs.

4. IQDEA*: Aggregation scheme that is similar in operations to IQDEA, with the following differences: (i) h^a_Δ is accurately estimated; and (ii) data from all activated nodes is collected at fusion center.

The fusion center v_0 is located near the bottom left hand corner of the terrain of size $\{600m \times 600m\}$. Sensor nodes are uniform-randomly distributed in the network. The transmission range (as well as sensing range) of each node is 60 meters, and performance results are averaged over 50 seed runs. The adaptive and anycast algorithms of A^2-MAC in Chapter 3 are used for duty cycle assignment and medium access control. Table 5.1 summarizes the simulation parameters used in the performance evaluation.

5.5.1 Varying Distance between PoI (Event) and Fusion Center

The distance between the PoI and v_0 in a network with 350 sensor nodes and average node degree of 10 neighbors is varied in Figure 5.13. The x/y coordinates of the PoI is increased from 250 meters to 450 meters[9]. The network diameter

[9]A distance of 250m to the PoI implies that the PoI is located at $\{250m \times 250m\}$.

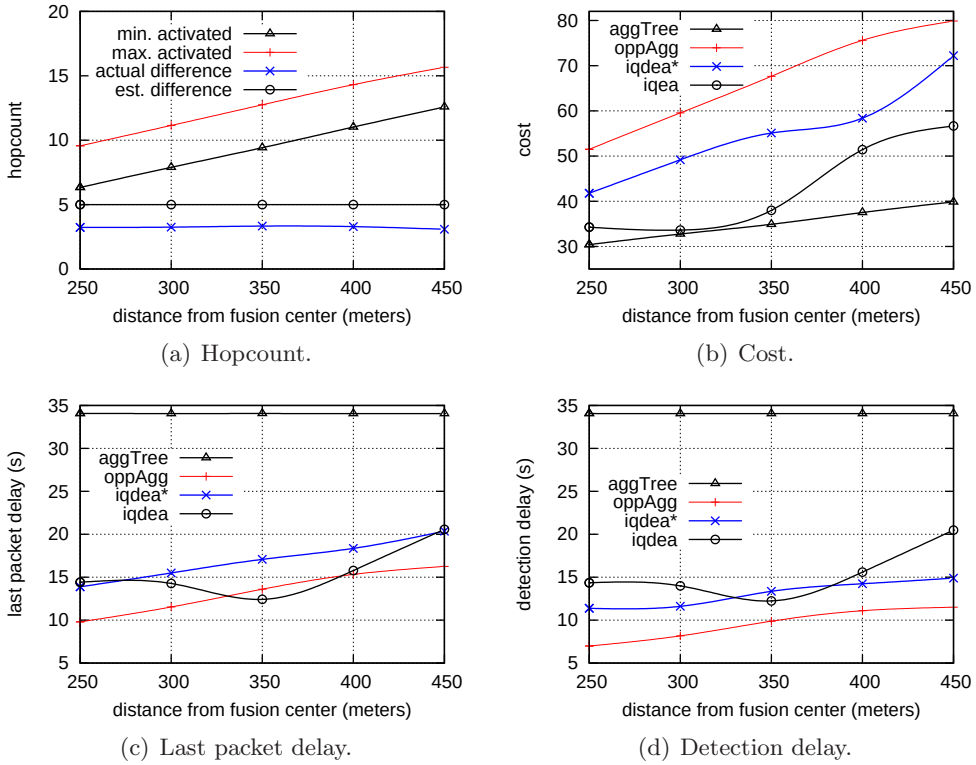
Table 5.1: Simulation Parameters

Parameter	Value
Terrain size	$\{600m \times 600m\}$
Network density	350 to 950 nodes
Transmission range	60 meters
Minimum active probability $\frac{\alpha_i}{n_c}$	0.01 (1 %)
MAC slot length τ	20 ms
Sensing interval	5 seconds
Signal strength s_0 (measured within distance r_a)	12.0
Minimum sensing range for signal attenuation r_a	15.0
Sensing capacity decay δ	0.02 to 0.05
Per-sample false alarm probability p_0	0.35
Targeted detection probability P_d	0.99
Targeted false alarm probability P_f	0.001 to 0.05

is approximately 18 hops. The minimum and maximum hopcounts (denoted as *min. activated* and *max. activated* respectively) of the activated region increase correspondingly as illustrated in Figure 5.13(a). The actual maximum difference (denoted as *actual difference*) and estimated maximum difference using Equation 5.15 (denoted as *est. difference*) remain constant as the diameter of the activated region does not vary with increasing distance between PoI and v_0[10].

The total cost (measured as total number of data transmissions) in Figure 5.13(b) increases with increasing distance between the PoI and v_0, due to the larger number of hops that data has to travel before reaching v_0. As data is maximally aggregated at each hop before transmission in AggTree, it incurs the least cost among all the aggregation schemes. In contrast, data aggregation occurs only opportunistically in OppAgg, leading to multiple forwarding paths between the activated region and v_0, and subsequently higher cost. By adapting the aggregation latency of nodes according to the (actual/estimated) hopcount difference between activated nodes, both IQDEA* and IQDEA incur less cost than

[10]The actual and estimated maximum differences in hopcounts of activated nodes are used in the computation of aggregation latencies of each node in IQDEA* and IQDEA respectively.



(a) Hopcount. (b) Cost.

(c) Last packet delay. (d) Detection delay.

Figure 5.13: Performance with increasing distance from PoI (event).

OppAgg. However, as up to two forwarding phases may be necessary in IQDEA* and IQDEA, they incur higher cost than AggTree. Furthermore, IQDEA can achieve better cost savings than IQDEA* as data acquisition terminates as soon as sufficient data has been collected for reliable PoI detection[11].

The aggregation latency of each node in AggTree is primarily dependent on its hopcount, network diameter and maximum per-hop delay. Consequently, its last packet delay (time taken for the last packet to reach fusion center) and detection delay (time taken for fusion center to detect PoI) in Figures 5.13(c) and 5.13(d) respectively, remain consistent despite the increasing distance of the PoI from the fusion center. As OppAgg does not incur any aggregation latency, its last packet delay and detection delay increase in proportion to the distance. The aggregation latency model used by IQDEA* and IQDEA is independent

[11] An average of 67% of the IQ from all the activated nodes is collected in IQDEA.

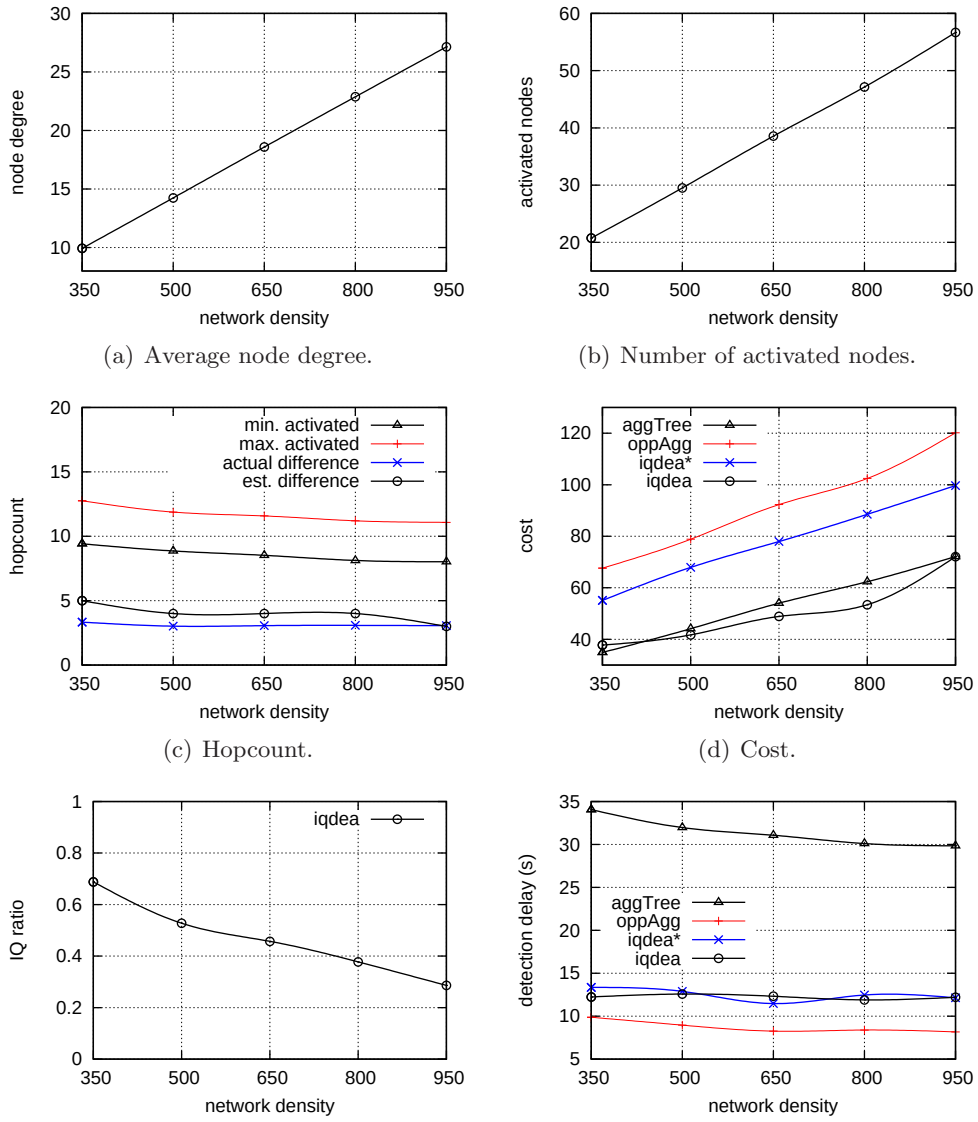
of absolute PoI location and network diameter; this results in significant delay savings as compared to *aggDelay*. Although IQDEA incurs some errors in its estimate of the maximum hopcount difference within the activated region, it can achieve comparable detection delay performance to IQDEA*.

5.5.2 Varying Network Density

Figure 5.14 studies the performance of the protocols as the network density is varied from 350 nodes to 950 nodes, yielding an average node degree from 10 to 27 neighbors as shown in Figure 5.14(a). As network density increases, the spatial distance between nodes decrease, leading to an increase in number of activated nodes in Figure 5.14(b). The PoI is located at $\{350m, 350m\}$ in the monitored terrain, such that the minimum and maximum activated hopcounts are approximately 8 hops and 11 hops respectively in Figure 5.14(c).

As the number of activated nodes increases in proportion to network density, the total cost incurred by all the protocols increase correspondingly in Figure 5.14(d), with OppAgg bounding the upper limit due to its opportunistic nature in data aggregation. When data is collected from all the activated nodes in the network, AggTree incurs the least cost as it maximizes aggregation opportunities at each hop along the forwarding path. Through exploitation of IQ awareness to suppress data when sufficient IQ has been acquired for reliable PoI detection, IQDEA is able to incur less cost than AggTree. The ratio of the IQ that is aggregated at the fusion center using IQDEA to total IQ generated at all the activated nodes decreases with increasing network density in Figure 5.14(e).

As the location of the PoI is static, the detection delay for all the aggregation schemes remain largely consistent in Figure 5.14(f), with AggTree and OppAgg bounding the upper and lower limits due to maximal and minimal aggregation opportunities respectively. Both IQDEA* and IQDEA are able to achieve low detection delays using bounded aggregation latencies at each node.



(a) Average node degree.

(b) Number of activated nodes.

(c) Hopcount.

(d) Cost.

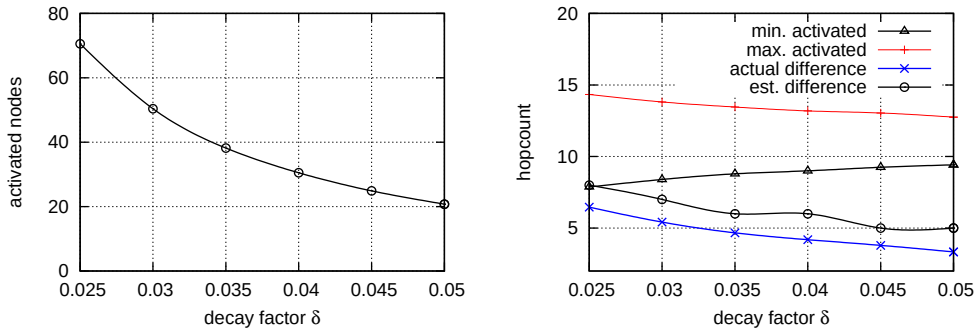
(e) IQ ratio.

(f) Detection delay.

Figure 5.14: Performance with increasing network density.

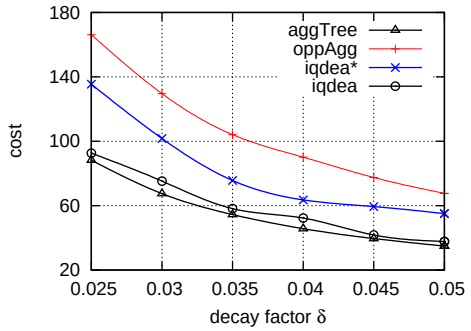
5.5.3 Varying Decay Factor δ

We vary the size of the activated region by increasing the decay factor δ of the sensing model in Equation 5.2 from 0.02 to 0.05 in Figure 5.15. There are 350 sensor nodes in the network and the PoI is statically located at $\{350m, 350m\}$ in the terrain. As δ increases, the number of activated nodes and hopcount difference in the activated region decrease correspondingly in Figures 5.15(a)

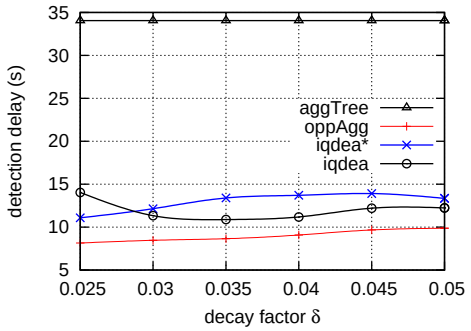


(a) Number of activated nodes.

(b) Hopcount.



(c) Cost.



(d) Detection delay.

Figure 5.15: Performance with increasing decay factor.

and 5.15(b) respectively.

Due to the reduction in number of activated nodes as δ increases, the total cost incurred by all the aggregation schemes decrease in Figure 5.15(c). OppAgg incurs the greatest cost as data is forwarded with minimal aggregation, while AggTree incurs the least cost as data is maximally aggregated at each hop towards the fusion center. IQDEA* and IQDEA reduce cost by aggregating data from nodes within each concentric circle before forwarding the aggregated data to the fusion center. The information quality awareness in IQDEA enables nodes to suppress data when sufficient data has been aggregated for reliable PoI detection, thereby further reducing the cost incurred by IQDEA*.

Due to the reduction in number of activated nodes and reduction in hopcount difference of activated nodes as δ increases, more nodes that are further away from the fusion center are required for PoI detection. This leads to slight in-

crease in detection delays of OppAgg, IQDEA* and IQDEA, which do not have rigid structures for data aggregation. However, the detection delay of AggTree remains somewhat constant despite the increase in δ in Figure 5.15(d), as its aggregation latency is independent of the characteristics of the activated region.

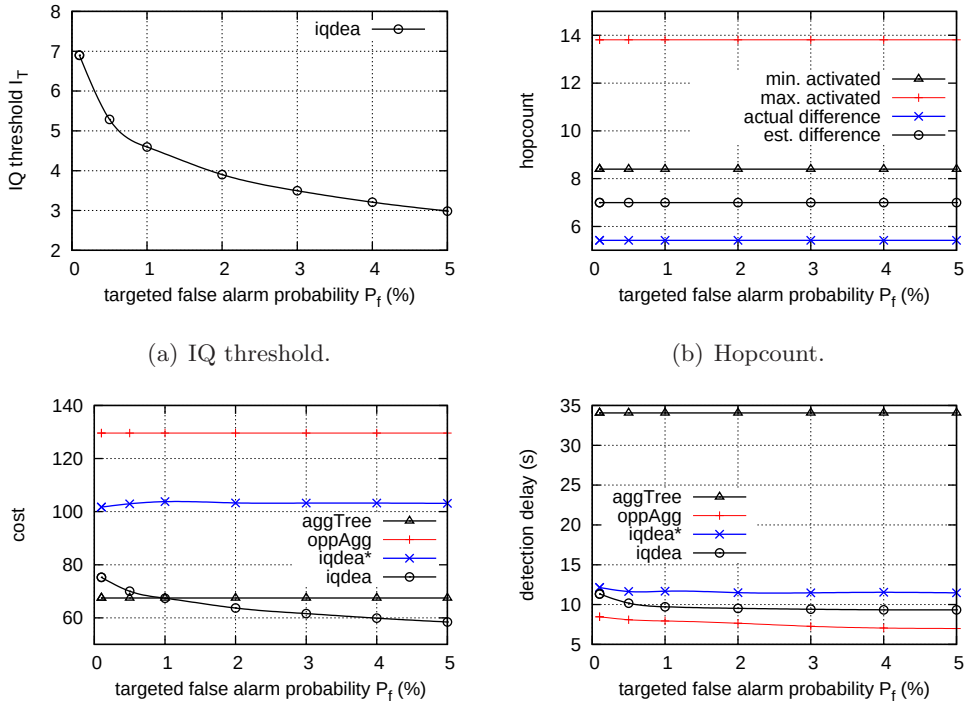
5.5.4 Varying Information Quality Threshold I_T

The IQ threshold I_T has an impact on the amount of data that is required for reliable PoI detection at the fusion center. Recall that that I_T is dependent on the targeted detection P_d and false alarm P_f probabilities. In Figure 5.16, we vary P_f from 0.1% to 5% in order to vary the IQ threshold for PoI detection. Figure 5.16(a) illustrates how the IQ threshold I_T decreases with increasing values of P_f. There are 350 nodes in the network and the PoI is located at $\{350m, 350m\}$ in the terrain.

Due to the static location of the PoI, the minimum and maximum hopcounts of activated nodes, as well as the maximum hopcount difference, remain constant with increasing values of P_f in Figure 5.16(b). As AggTree, OppAgg and IQDEA* acquire data from all the activated nodes in the network, the total costs incurred by these aggregation schemes in Figure 5.16(c) do not vary significantly with increasing P_f. In IQDEA, data acquisition terminates as soon as sufficient data has been collected for PoI detection. Subsequently, less data is required and total cost can be reduced by approximately 60% (from 92 to 38 transmissions) when P_f increases from 0.01% to 5%. There is only slight variation in the detection delays of the aggregation schemes as P_f increases in Figure 5.16(d) as the PoI is statically located and maximum hopcount difference between the activated nodes is very small.

5.5.5 Varying Errors in Hopcount Difference Estimation

We observe that the estimation of maximum hopcount difference h^a_Δ in the activated region computed using Equation 5.15 and used in IQDEA is often a



(a) IQ threshold.

(b) Hopcount.

(c) Cost.

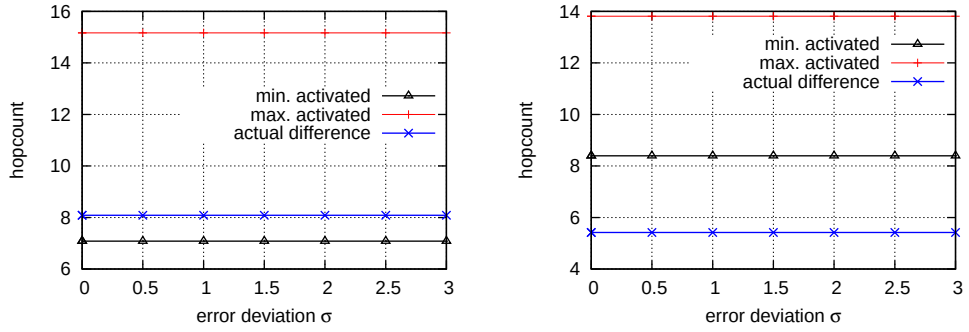
(d) Detection delay.

Figure 5.16: Performance with increasing targeted false alarm probability P_f.

deviation of the actual maximum hopcount difference. Figure 5.17 studies the performance of IQDEA as the error deviation σ of this maximum hopcount estimation[12] increases from 0 to 3. The network density is 350 nodes and the PoI is located at $\{350m, 350m\}$ in the terrain. The decay factor δ in Equation 5.2 - which influences the size of the activated region - is also varied; we present results of $\delta = \{0.02, 0.03\}$.

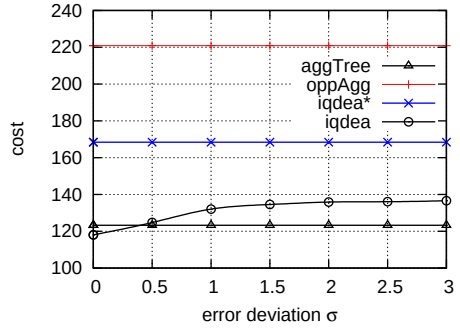
Figures 5.17(a) and 5.17(b) illustrate the minimum and maximum hopcounts of the activated regions with $\delta = 0.02$ and $\delta = 0.03$ respectively. With a smaller decay factor of $\delta = 0.02$, a larger region of the terrain is activated, leading to a larger average hopcount difference of 8 hops and approximately 100 activated nodes; the average hopcount difference when $\delta = 0.03$ is 5.4 hops with approximately 50 activated nodes.

[12]Note that each node may estimate the maximum hopcount difference with a different absolute error.

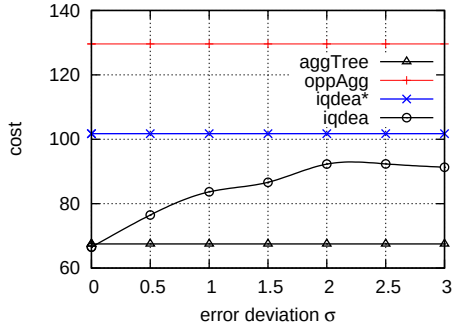


(a) Hopcount ($\delta = 0.02$).

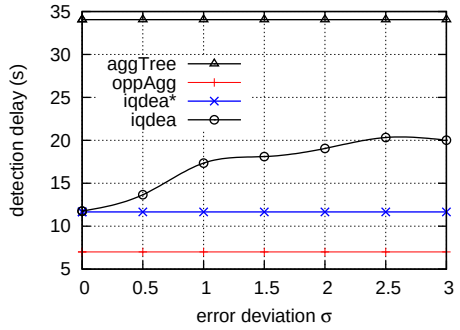
(b) Hopcount ($\delta = 0.03$).



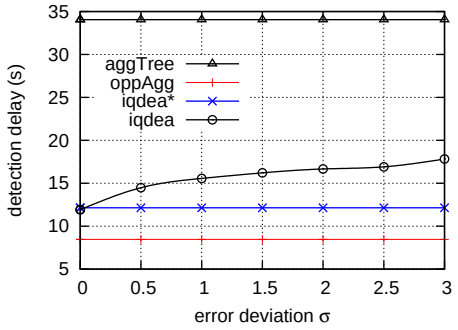
(c) Cost ($\delta = 0.02$).



(d) Cost ($\delta = 0.03$).



(e) Detection delay ($\delta = 0.02$).



(f) Detection delay ($\delta = 0.03$).

Figure 5.17: Performance with increasing error standard deviation.

As error deviation σ increases, the variance in hopcount difference estimations of each node increases. The aggregation latency within each concentric circle in IQDEA may no longer decrease with increasing distance from the fusion center. This reduces aggregation opportunities in IQDEA, leading to increased transmission costs in Figures 5.17(c) and 5.17(d). Although IQDEA* uses accurate hopcount difference in the computation of aggregation latency at each node, it incurs higher costs than IQDEA as the latter terminates data ac-

quisition when sufficient data has been collected for PoI detection. A smaller percentage increase of approximately 15% in cost as σ increases from 0 to 3 is observed with $\delta = 0.02$ as compared to the approximately 35% increase in cost with $\delta = 0.03$, as: (i) ratio of error deviation to actual hopcount difference is smaller with $\delta = 0.02$; and (ii) presence of greater number of activated nodes when $\delta = 0.02$ reduces the impact of errors in hopcount estimations as data from only a small proportion of activated are required at the fusion center.

The detection delays of the various aggregation schemes with varying σ values are presented in Figures 5.17(e) and 5.17(f). The increasing deviation of errors in maximum hopcount difference estimations lead to increase in detection delays of up to 70% and 50% with $\delta = 0.02$ and $\delta = 0.03$ respectively, when $\sigma = 3$. Despite this increase, IQDEA generally incurs less than 50% of the delays incurred by AggTree, thus exhibiting robustness to errors in the estimates.

5.6 Summary

This chapter presents and evaluates IQDEA - an *I*nformation *Q*uality aware *D*elay *E*fficient *A*ggregation scheme for energy constrained wireless sensor networks that are deployed for PoI detection. Existing energy efficient schemes that perform data aggregation to minimize transmission costs tend to incur extremely long detection latencies due to lack of knowledge of *a priori* PoI location.

In IQDEA, a novel function is proposed to compute the aggregation latency of each node based on the estimated maximum hopcount difference of the activated region; this eliminates the dependency of the aggregation latency on network diameter and absolute PoI location. Coupled with IQ-awareness to terminate data acquisition as soon as sufficient data has been collected for accurate and reliable PoI detection, IQDEA achieves a good tradeoff between energy and delay efficiency while satisfying IQ constraints. Through simulations, we validate that IQDEA incurs significantly less energy cost and detection delay than

opportunistic aggregation tree and structured aggregation tree respectively.

Chapter 6

Conclusion

In this chapter, we summarize the key research contributions and insights of this dissertation, followed by discussions on open issues and future work.

6.1 Key Research Contributions

One of the primary factors that inhibit the successful deployment of Wireless Sensor Networks is the severe energy constraints faced by the tiny computing and sensing devices which form the building blocks of these networks. Although sensor network applications are expected to have a lifetime of several years, commonly used sensor motes (from the Crossbow family) are powered by AA batteries, which severely limits their energy sources.

Among the various operations through which energy is expended in a sensor network, the bulk of energy expenditure is consumed by inter-nodal communications. We have identified the shortcomings of existing energy efficient communication protocols in Chapter 2, and provided suggestions on how they can be improved upon, via techniques such as: (i) leveraging on sensor network characteristics; (ii) incorporating IQ-awareness; (iii) incorporating cross-layer interactions; and (iv) adapting to network characteristics.

This dissertation therefore focuses on the design and development of commu-

nication protocols in energy constrained networks, in order to maximize energy efficiency and hence prolong network lifetime, without overly compromising on other performance metrics that are of interest to the application. Our main contributions are three novel energy efficient communication protocols that address the caveats of existing protocols and achieve good energy delay tradeoffs in energy constrained networks: (i) **A^2-MAC** [33] - *A*daptive, *A*nycast MAC protocol; (ii) **IQAR** [34] - *I*nformation *Q*uality *A*ware *R*outing protocol; and (iii) **IQDEA** - *I*nformation *Q*uality aware *D*elay *E*fficient *A*ggregation scheme.

6.1.1 A^2-MAC

A^2-MAC is an asynchronous and adaptive Medium Access Control (MAC) protocol that aims to arbitrate access to energy constrained nodes in a shared wireless medium, in an energy efficient and delay efficient manner. It utilizes: (i) random wakeup schedules; (ii) adaptive duty cycles; and (iii) adaptive anycast forwarder selection.

In large scale sensor networks that may span multiple hops, it is costly and impractical to provide fine-grained time synchronization to all the nodes. With the use of random wakeup schedules, each node can independently and randomly select its wakeup schedule without coordination and time synchronization.

Existing MAC protocols typically assign the same (high) duty cycle to each node in the network; thus, all nodes will fail from energy drain at about the same time, resulting in premature termination of the usefulness of the network. A^2-MAC allows each node to adopt a different duty cycle based on its local network topology. As sensor networks are typically densely deployed with sufficient node redundancy, this enables the network to fail gracefully over time although some nodes may fail earlier than the rest of the network.

With duty cycling, long sleep latencies may be incurred in waiting for a potential forwarding node to wake up. Through adaptive anycast selection of forwarding nodes, A^2-MAC can significantly reduce end-to-end delays and provide

robustness to intermittent connectivity which is inherent of wireless networks.

Our performance evaluation of A^2-MAC verifies the significance of utilizing adaptive and anycast forwarder selection, particularly when duty cycles are low and in the presence of intermittent link connectivity. Despite the use of only local topological information in duty cycle adaptation and selection, A^2-MAC allows the network to fail gracefully over time and can provide longer connectivity as compared to schemes that assign the same duty cycles to every node. As compared to existing asynchronous MAC protocols, A^2-MAC can achieve better connectivity, higher coverage, as well as lower energy consumption and delays.

6.1.2 IQAR

IQAR is an information quality aware routing protocol that aims to find a least cost (minimum energy) routing tree that satisfies a given IQ constraint within a delay bound. It utilizes: (i) topology-aware histogram-based aggregation structure that encapsulates the cost of including the IQ contribution of each activated node in a compact and efficient way; and (ii) greedy heuristic to approximate and prune a least cost aggregation routing path.

When a phenomenon of interest (PoI) occurs, multiple sensor nodes may be activated in a typically densely deployed sensor network, leading to data implosion and redundancy. The optimal least cost routing solution that allows the PoI to be detected with sufficient reliability and accuracy at the fusion center, is a variation of the classical NP-hard Steiner tree problem in graphs - and requires knowledge of the global network topology as well as individual IQ contributions of each activated sensor node.

In IQAR, we propose a topology-aware histogram-based aggregation structure that can encapsulate the IQ contributions of each activated node in a compact and efficient manner. This greatly reduces the computational costs incurred in acquiring global network information at the fusion center. Upon obtaining an approximate view of the topology and IQ contributions of the activated node,

the fusion center then uses a greedy heuristic to approximate and prune a least-cost aggregation routing path. Redundant data is suppressed for a time interval to reduce traffic load and alleviate medium access contention.

Through simulations, we show that IQAR can achieve significant energy and delay savings while maintaining information quality in event detection.

6.1.3 IQDEA

IQDEA is an information quality aware delay efficient aggregation scheme for energy constrained wireless sensor networks that are deployed for PoI detection. It aims to minimize the PoI detection delay in mission critical applications, without compromising on energy efficiency. The information quality of data provided by each activated node is used to determine: (i) aggregation latency; and (ii) forwarding decision at each intermediate forwarding node.

In aggregation schemes, each node waits for an *aggregation latency* before forwarding its data to the fusion center. This allows data from its upstream nodes to be aggregated together before the forwarding process, thus minimizing the energy costs that are incurred during data acquisition. However, most aggregation schemes ignore the existence of energy-delay tradeoffs and incur extremely long delays, especially in a duty cycled network where nodes are asleep for most of the time and when PoI locations are not known *a priori*.

IQDEA minimizes the aggregation latency of each node by introducing a novel aggregation latency function that is independent of absolute PoI location and network diameter. Nodes are grouped into concentric circles centered at the fusion center, and the aggregation latency is computed based on the relative position of nodes in each circle as well as estimated hopcount difference in the activated region. IQDEA exploits IQ awareness to: (i) minimize PoI latency by forwarding data to fusion center immediately without additional aggregation latency; and (ii) minimize energy expenditure by terminating data acquisition as soon as sufficient data has been collected for reliable PoI detection. To further

minimize PoI detection delay, the forwarding node is dynamically selected based on the instantaneous expected end-to-end delay towards the fusion center.

Through our performance evaluation, we show that IQDEA achieves a good tradeoff between energy and delay efficiency. In particular, it incurs significantly less cost and detection latencies than structureless (opportunistic) aggregation and structured aggregation respectively.

6.2 Insights

The following summarizes the key insights gained from the research work in this dissertation.

1. **Adaptation is necessary in non-uniform and/or dynamic environments.** Many existing protocols are designed with uniform and/or static parameters in mind. However, the real environment is often full of dynamics, and conforms to neither regularity nor uniformity. A typical example is that of the inherent time-varying wireless link characteristics and varying node degree of each node in the network. Without adaptation, nodes cannot make use of the resources in an efficient manner. The key distinction between A^2-MAC and other MAC protocols is its adaptive behavior towards local node topology, thus allowing it to reduce the duty cycles of each node whenever possible. In both IQAR and IQDEA, routes are adapted based on PoI location and current level of information quality at each node.

2. **Diversity can improve performance.** Due to the unreliability of the wireless channel and non-uniformity of the network, different channels and routes display varying characteristics at different times. Intelligent exploitation of both channel and route diversity can lead to potential improvements in network performance. For example, the use of an anycast

mechanism in A^2-MAC allows nodes to exploit diversity by selecting any awake nodes in their forwarding sets as the next hops, and thus minimize end-to-end latencies.

3. **Information quality can be used to reduce the amount of data acquisition.** Sensory data tends to exhibit spatio-temporal correlation, especially in dense sensor network deployments. Conventional sensor network protocols assume that all generated sensory data is required at the fusion center; however, it is often the case that this form of complete data acquisition is not only unnecessary but also incurs excessive overheads. By pegging an information quality level to each sensory data, sensor network algorithms can thus evaluate if the amount of aggregated data is sufficient for post-processing at the fusion center. This reduces the amount of data acquisition and increase the efficiency of resource utilization, thereby improving network and application-level performance.

4. **Topology-aware hints can provide suggestions for better route establishment.** While availability of global topological information enables 'optimal' routes to be established, the acquisition of such knowledge incurs significant overheads and is not be practical in dynamic environments. However, it is possible to provide topology-aware hints through a compact histogram, as in the case of IQAR. These hints are small enough to be piggybacked onto data packets, incur minimal overheads, and enable better route establishment.

6.3 Open Issues and Future Work

The last decade has seen rapid proliferation of tiny computing and sensing devices that are integrated into our daily lives via a wide spectrum of applications. While these devices have increasingly sophisticated microprocessors as well as

superior processing and sensing capabilities, energy expenditure remains a major cause of concern as portable energy sources have limited capacities.

This dissertation focuses on the communication aspect of protocol design in energy constrained networks and proposes three novel energy efficient protocols - A^2-MAC, IQAR and IQDEA, with the latter two being particularly designed for event driven sensor networks, and A^2-MAC being applicable to generic wireless sensor networks. In the following, we outline some of the open issues related to these three protocols and then discuss some directions for future work.

The adaptation of duty cycles in A^2-MAC is currently based only on local node topology. However, it is well known that: (i) nodes in WSNs have varying traffic demands; and (ii) nodes that are nearer to the fusion center tend to have higher data traffic and hence expend more energy than the rest of the nodes in the network. One possible extension of A^2-MAC is to incorporate both traffic demand and some form of global topological information into the adaptation.

There are a few simplifying assumptions made in the sensing models used in IQAR and IQDEA that may not apply across all classes of PoIs.

1. The signal strength associated with each observed signal monotonically decreases with increasing distance from the PoI. While this model holds true for most PoIs (such as light intensity, salinity, pressure and sound), this assumption may not apply for other PoIs (such as motion and velocity). Furthermore, it is not trivial to determine each of the parameters in the sensing model. It is worthwhile investigating how well the proposed aggregation schemes will perform when the sensing model cannot fit the actual sensing characteristics perfectly, and what types of modifications (if any) are required to the aggregation schemes in such scenarios.

2. Each sensor reading is assumed to be independent and identically distributed (i.i.d.) across sensor nodes. However, it is often the case that sensory data is spatially correlated, i.e. sensor nodes within the same ge-

ographical vicinity tend to sense similar data. By taking such correlations into account, anomalies in sensory data can be quickly detected. This can lead to improvements in resource utilization and detection accuracy of the entire system.

In general, each sensor network may also be deployed for the detection of multiple PoIs. We have not yet evaluated the performance of IQAR and IQDEA when multiple PoIs occur simultaneously at various locations within the same network. The current performance studies of our proposed protocols are based on simulations in network simulators such as GloMoSim and Qualnet. As part of future work, one can implement these protocols on sensor mote testbeds to evaluate their performance under more severe and realistic conditions.

On a broader scale, the design space for energy efficient communication protocols is still very large, and there exists room for continued research efforts in the following:

1. **An integrated framework for energy efficient communication protocols.** Existing communication protocols for energy constrained networks are mostly designed for a specific layer in the networking protocol stack. Cross layer interactions are often ignored, despite increasing evidence that the exploitation of information from lower level networking layers can vastly improve system performance. At the current moment, there exists lack of a comprehensive framework that succinctly integrates energy efficient protocols from the various layers in the networking protocol stack together.

2. **Energy efficient protocols for the evolving hardware.** As MEMS technology advances, it is likely that the various radio hardware models that are popularly used today (such as CC2420 in the MicaZ and TelosB motes) will become obsolete and be replaced by new hardware with vastly different operating characteristics. While the SLEEP mode in new radios

is still expected to consume significantly less energy than the other modes, it is plausible that: (i) energy consumption levels of the transmitting (TX) and/or receiving (RX) modes may be of some orders of magnitude lower than it is today; and/or (ii) relative difference in energy consumption levels of TX and RX mode may either increase or decrease. In either case, it may be necessary to relook at existing protocols, which are predominantly transmitter-oriented, and evaluate if there is a need to modify their operational methodologies.

3. **Energy efficient protocols for ubiquitous sensors.** With the increasing penetration rates and ability of smartphones to provide various sensing capabilities, it is likely that sensors of the future will become even more ubiquitous than it is today. In contrast with the state-of-the-art sensors that are large static in nature, smartphone sensors are mobile with possibly predictable mobility patterns. Communication between the smartphone sensors can take place either through: (i) short-range Bluetooth or WiFi connectivity; and/or (ii) long-range 3G connectivity. Disconnections in link connectivity can also be attributed to node mobility in addition to duty cycling and unreliability of the wireless channel. A new suite of protocols may then be necessary to provide energy efficiency in this new class of sensors.

Bibliography

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: a Survey," *Computer Networks*, vol. 38, no. 4, 2002.

[2] S. Tilak and N. B. Abu-Ghazaleh, "A Taxonomy of Wireless Micro-Sensor Network Models," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6, no. 2, 2002.

[3] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet," in *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X)*, 2002.

[4] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," in *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, 2002.

[5] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An Analysis of a Large Scale Habitat Monitoring Application," in *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2004.

[6] D. J. Abadi, S. Madden, and W. Lindner, "REED: Robust, Efficient Filtering and Event Detection in Sensor Networks," in *International Conference on Very Large Data Bases (VLDB)*, 2005.

[7] P. Dutta, M. Grimmer, A. Arora, S. Bibyk, and D. Culler, "Design of a Wireless Sensor Network Platform for Detecting Rare, Random, and Ephemeral events," in *International Symposium on Information Processing in Sensor Networks (IPSN)*, 2005.

[8] K. Casey, A. Lim, and G. Dozier, "A Sensor Network Architecture for Tsunami Detection and Response," *International Journal of Distributed Sensor Networks*, vol. 4, no. 1, 2008.

[9] P. Bonnet, J. Gehrke, and P. Seshadri, "Querying the Physical World," *IEEE Personal Communications*, vol. 7, 2000.

[10] M. Hefeeda and M. Bagheri, "Forest Fire Modeling and Early Detection using Wireless Sensor Networks," *Ad Hoc and Sensor Wireless Networks*, vol. 7, 2009.

[11] G. Coyle, L. Boydell, and L. Brown, "Home Telecare for the Elderly," *Journal of Telemedicine and Telecare*, vol. 1, no. 3, 1995.

[12] A. Woo, T. Tong, and D. Culler, "Taming the Underlying Challenges of Multihop Routing in Sensor Networks," in *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2003.

[13] J. Zhao and R. Govindan, "Understanding Packet Delivery Performance in Dense Wireless Sensor Networks," in *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2003.

[14] S. Rajasegarar, J. Gubbi, O. Bondarenko, S. Kininmonth, S. Marusic, S. Bainbridge, I. Atkinson, and M. Palaniswami, "Sensor Network Implementation Challenges in the Great Barrier Reef Marine Environment," in *ICT Mobile and Wireless Communications Summit (ICT-MobileSummit)*, 2008.

[15] D. Culler, D. Estrin, and M. Srivastava, "Overview of Sensor Networks," *Computer*, vol. 37, no. 8, 2004.

[16] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "Complex Behavior at Scale: An Experimental Study of Low-Power Wireless Sensor Networks," UCLA Computer Science Technical Report UCLA/CSD-TR 02-0013, Tech. Rep., 2002.

[17] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-Aware Wireless Microsensor Networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, 2002.

[18] "Crossbow technology." [Online]. Available: http://www.xbow.com

[19] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen, "A Survey of Energy Efficient Network Protocols for Wireless Networks," *Wireless Networks*, vol. 7, no. 4, 2001.

[20] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy Conservation in Wireless Sensor Networks: a Survey," *Ad Hoc Networks*, vol. 7, no. 3, 2009.

[21] W. Ye and J. Heidemann, "Medium Access Control in Wireless Sensor Networks," in *Wireless Sensor Networks*, T. Znati, K. Sivalingam, and C. Raghavendra, Eds. KAP, 2004, pp. 73–91.

[22] P. Naik and K. Sivalingam, "A Survey of MAC Protocols for Sensor Networks," in *Wireless Sensor Networks*, T. Znati, K. Sivalingam, and C. Raghavendra, Eds. KAP, 2004, pp. 93–107.

[23] G. P. Halkes, T. van Dam, and K. G. Langendoen, "Comparing Energy-Saving MAC Protocols for Wireless Sensor Networks," *Mobile Networks and Applications*, vol. 10, no. 5, 2005.

[24] K. Langendoen and G. P. Halkes, "Energy-Efficient Medium Access Control," in *Embedded Systems Handbook*, R. Zurawski, Ed. CRC press, 2005, pp. 34.1–34.29.

[25] I. Demirkol, C. Ersoy, and F. Alagoz, "Mac Protocols for Wireless Sensor Networks: a Survey," *IEEE Communications Magazine*, vol. 44, no. 4, 2006.

[26] K. Kredo II and P. Mohapatra, "Medium Access Control in Wireless Sensor Networks," *Computer Networks*, vol. 51, no. 4, 2007.

[27] J. N. Al-Karaki and A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey," *IEEE Wireless Communications*, vol. 11, no. 6, 2004.

[28] K. Akkaya and M. Younis, "A Survey on Routing Protocols for Wireless Sensor Networks," *Ad Hoc Networks*, vol. 3, no. 3, 2005.

[29] "IEEE 802.11, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE Standard, 1999.

[30] D. Qiao, S. Choi, A. Soomro, and K. G. Shin, "Energy Efficient PCF Operation of IEEE 802.11a Wireless LAN," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2002.

[31] C. E. Perkins, *Ad Hoc Networking*. Addison-Wesley, 2001.

[32] X. Xia, Q. Ren, and Q. Liang, "Cross-Layer Design for Mobile Ad Hoc Networks: Energy, Throughput and Delay-Aware Approach," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2006.

[33] H. X. Tan and M. C. Chan, "A^2-MAC: An Adaptive, Anycast MAC Protocol for Wireless Sensor Networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2010.

[34] H. X. Tan, M. C. Chan, W. Xiao, P. Y. Kong, and C. K. Tham, "Information-Quality Aware Routing in Event-Driven Sensor Networks," in *IEEE Conference on Computer Communications (INFOCOM)*, 2010.

[35] B. Krishnamachari, D. Estrin, and S. Wicker, "Modeling Data-Centric Routing in Wireless Sensor Networks," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2002.

[36] Y. Gu, T. He, M. Lin, and J. Xu, "Spatiotemporal Delay Control for Low-Duty-Cycle Sensor Networks," in *IEEE Real-Time Systems Symposium (RTSS)*, 2009.

[37] R. C. Shah and J. M. Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2002.

[38] I. Kang and R. Poovendran, "Maximizing Static Network Lifetime of Wireless Broadcast Ad Hoc Networks," in *IEEE International Conference on Communications (ICC)*, 2003.

[39] H. Kwon, T. H. Kim, S. Choi, and B. G. Lee, "Cross-layer Lifetime Maximization under Reliability and Stability Constraints in Wireless Sensor Networks," in *IEEE International Conference on Communications (ICC)*, 2005.

[40] T. L. Lim and G. Mohan, "Energy Aware Geographical Routing and Topology Control to Improve Network Lifetime in Wireless Sensor Networks," in *International Conference on Broadband Networks*, 2005.

[41] Y. Cui, Y. Xue, and K. Nahrstedt, "A Utility-based Distributed Maximum Lifetime Routing Algorithm for Wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 3, 2006.

[42] M. Maleki, K. Dantu, and M. Pedram, "Lifetime Prediction Routing in Mobile Ad Hoc Networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2003.

[43] A. Alfieri, A. Bianco, P. Brandimarte, and C. F. Chiasserini, "Exploiting Sensor Spatial Redundancy to Improve Network Lifetime," in *IEEE Global Telecommunications Conference (GLOBECOM)*, 2004.

[44] T. Suzuki, M. Bandai, and T. Watanabe, "Dispersivecast: Dispersive Packets Transmission to Multiple Sinks for Energy Saving in Sensor Networks," in *International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2006.

[45] X. Tang and J. Xu, "Extending Network Lifetime for Precision-Constrained Data Aggregation in Wireless Sensor Networks," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2006.

[46] "Texas instruments." [Online]. Available: http://www.ti.com/

[47] E. Shih, S. H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks," in *ACM International Conference on Mobile Computing and Networking (MOBICOM)*, 2001.

[48] P. Santi, "Topology Control in Wireless Ad Hoc and Sensor Networks," *ACM Computing Survey*, vol. 37, no. 2, 2005.

[49] M. Burkhart, P. von Rickenbach, R. Wattenhofer, and A. Zollinger, "Does Topology Control Reduce Interference?" in *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2004.

[50] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2002.

[51] E. Shih, P. Bahl, and M. J. Sinclair, "Wake on Wireless: an Event Driven Energy Saving Strategy for Battery Operated Devices," in *ACM International Conference on Mobile Computing and Networking (MOBICOM)*, 2002.

[52] R. Bahl, A. Adya, J. Padhye, and A. Walman, "Reconsidering Wireless Systems with Multiple Radios," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 5, 2004.

[53] W. Ye, J. Heidemann, and D. Estrin, "Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks," *IEEE ACM Transactions on Networking*, vol. 12, no. 3, 2004.

[54] T. van Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2003.

[55] G. Lu, B. Krishnamachari, and C. Raghavendra, "An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Sensor Networks," in *International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN)*, 2004.

[56] S. Du, A. K. Saha, and D. B. Johnson, "RMAC: A Routing-Enhanced Duty-Cycle MAC Protocol for Wireless Sensor Networks," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2007.

[57] J. Polastre, J. Hill, and D. E. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," in *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2004.

[58] M. Buettner, G. Yee, E. Anderson, and R. Han, "X-MAC: A Short Preamble MAC Protocol For Duty-Cycled Wireless Networks," in *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2006.

[59] S. Liu, K. W. Fan, and P. Sinha, "CMAC: An Energy Efficient MAC Layer Protocol Using Convergent Packet Forwarding for Wireless Sensor Networks," in *IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2007.

[60] S. Doshi and T. X. Brown, "Minimum Energy Routing Schemes for a Wireless Ad Hoc Network," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2002.

[61] A. Srinivas and E. Modiano, "Minimum Energy Disjoint Path Routing in Wireless Ad-Hoc Networks," in *ACM International Conference on Mobile Computing and Networking (MOBICOM)*, 2003.

[62] J. Zhu, C. Qiao, and X. Wang, "A Comprehensive Minimum Energy Routing Scheme for Wireless Ad Hoc Networks," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2004.

[63] N. Park, D. Kim, Y. Doh, S. Lee, and J. T. Kim, "An Optimal and Lightweight Routing for Minimum Energy Consumption in Wireless Sensor Networks," in *IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2005.

[64] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks," *Mobile Computing and Communications Review*, vol. 1, no. 2, 2002.

[65] Y. Chen and N. Nasser, "Energy-Balancing Multipath Routing Protocol for Wireless Sensor Networks," in *International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine)*, 2006.

[66] N. Jain, D. K. Madathil, and D. P. Agrawal, "MidHopRoute: a Multiple Path Routing Framework for Load Balancing with Service Differentiation

in Wireless Sensor Networks," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 1, no. 4, 2006.

[67] C. Y. Wan, A. T. Campbell, and L. Krishnamurthy, "Pump Slowly, Fetch Quickly (PSFQ): a Reliable Transport Protocol for Sensor Networks," in *IEEE Journal on Selected Areas in Communications*, 2005.

[68] Özgür B. Akan and I. F. Akyildiz, "Event-to-Sink Reliable Transport in Wireless Sensor Networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 13, no. 5, 2005.

[69] Y. Wang, C. Y. Wan, M. Martonosi, and L. S. Peh, "Transport Layer Approaches for Improving Idle Energy in Challenged Sensor Networks," in *SIGCOMM Workshop on Challenged Networks*, 2006.

[70] B. Krishnamachari, D. Estrin, and S. Wicker, "The Impact of Data Aggregation in Wireless Sensor Networks," in *ICDCS Workshops*, 2002.

[71] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-network Aggregation Techniques for Wireless Sensor Networks: A Survey," *IEEE Wireless Communications*, vol. 14, no. 2, 2007.

[72] R. M. Karp, "Reducibility Among Combinatorial Problems," in *Symposium on the Complexity of Computer Computations*, 1972.

[73] H. Du, X. Hu, and X. Jia, "Energy Efficient Routing and Scheduling for Real-Time Data Aggregation in WSNs," *Computer Communications*, vol. 29, no. 17, 2006.

[74] H. Luo, Y. Liu, and S. K. Das, "Routing Correlated Data with Fusion Cost in Wireless Sensor Networks," *IEEE Transactions on Mobile Computing (TMC)*, vol. 5, no. 11, 2006.

[75] Y. Yu, V. K. Prasanna, and B. Krishnamachari, "Energy Minimization for Real-Time Data Gathering in Wireless Sensor Networks," *Transactions on Wireless Communications*, vol. 5, no. 11, 2006.

[76] H. Lee and A. Keshavarzian, "Towards Energy-optimal and Reliable Data Collection via Collision-free Scheduling in Wireless Sensor Networks," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2008.

[77] K. Yuen, B. Liang, and B. Li, "A Distributed Framework for Correlated Data Gathering in Sensor Networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 1, 2008.

[78] Y. Wu, Y. H. Liu, and W. Lou, "Energy-Efficient Wake-Up Scheduling for Data Collection and Aggregation," *TPDS*, vol. 21, no. 2, 2010.

[79] S. Gandham, Y. Zhang, and Q. Huang, "Distributed Minimal Time Convergent Scheduling in Wireless Sensor Networks," in *ICDCS*, 2006.

[80] P. J. Wan, S. C. H. Huang, L. Wang, Z. Wan, and X. Jia, "Minimum-Latency Aggregation Scheduling in Multihop Wireless Networks," in *MobiHoc*, 2009.

[81] B. Yu, J. Li, and Y. Li, "Distributed Data Aggregation Scheduling in Wireless Sensor Networks," in *IEEE Conference on Computer Communications (INFOCOM)*, 2009.

[82] M. O. Diaz and K. K. Leung, "Randomized Scheduling Algorithm for Data Aggregation in Wireless Sensor Networks," in *European Wireless Conference*, 2010.

[83] C. Joo, J. G. Choi, and N. B. Shroff, "Delay Performance of Scheduling with Data Aggregation in Wireless Sensor Networks," in *INFOCOM*, 2010.

[84] Y. Li, L. Guo, and S. K. Prasad, "An Energy-Efficient Distributed Algorithm for Minimum-Latency Aggregation Scheduling in Wireless Sensor Networks," in *ICDCS*, 2010.

[85] X. Xu, X. Y. Li, X. Mao, S. Tang, and S. Wang, "A Delay-Efficient Algorithm for Data Aggregation in Multihop Wireless Sensor Networks," *TPDS*, 2010.

[86] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," in *Hawaii International Confereence on System Sciences*, 2000.

[87] ——, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *Transactions in Wireless Communications*, vol. 1, no. 4, 2002.

[88] S. Lindsey and C. S. Raghavendra, "PEGASIS: Power-Efficient Gathering in Sensor Information Systems," in *Aerospace Conference*, 2002.

[89] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita, "A Line in the Sand: A Wireless Sensor Network for Target Detection, Classification, and Tracking," *Computer Networks*, vol. 46, no. 5, 2004.

[90] B. M. Sullivan, "Bioterrorism Detection: The Smoke Alarm and the Canary," *Technology Review Journal*, vol. 11, no. 1, 2003.

[91] K. W. Fan, S. Liu, and P. Sinha, "On the Potential of Structure-Free Data Aggregation in Sensor Networks," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2006.

[92] ——, "Structure-Free Data Aggregation in Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 8, 2007.

[93] ——, "Scalable Data Aggregation for Dynamic Events in Sensor Networks," in *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2006.

[94] K. W. Fan and P. Sinha, "Distributed Online Data Aggregation for Large Scale Sensor Networks," in *MASS*, 2008.

[95] Y. A. Oswald, S. Schmid, and R. Wattenhofer, "Tight Bounds for Delay-Sensitive Aggregation," in *ACM Symposium on Principles of Distributed Computing (PODC)*, 2008.

[96] C. Y. Wan, S. B. Eisenman, A. T. Campbell, and J. Crowcroft, "Siphon: Overload Traffic Management using Multi-Radio Virtual Sinks in Sensor Networks," in *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2005.

[97] E. I. Oyman and C. Ersoy, "Multiple Sink Network Design Problem in Large Scale Wireless Sensor Networks," in *IEEE International Conference on Communications (ICC)*, 2004.

[98] P. Ciciriello, L. Mottola, and G. P. Picco, "Efficient Routing from Multiple Sources to Multiple Sinks in Wireless Sensor Networks," in *European Conference on Wireless Sensor Networks (EWSN)*, 2007.

[99] Z. Vincze, R. Vida, and A. Vidacs, "Deploying Multiple Sinks in Multi-hop Wireless Sensor Networks," in *IEEE International Conference on Pervasive Services (ICPS)*, 2007.

[100] P. McDermott-Wells, "What is Bluetooth?" *IEEE Potentials*, vol. 23, no. 5, 2005.

[101] S. Mattisson, "Low-Power Considerations in the Design of Bluetooth," in *International Symposium on Low Power Electronics and Design (ISLPED)*, 2000.

[102] O. Karjalainen, S. Rantala, and M. Kivikoski, "A Comparison of Bluetooth Low Power Modes," in *International Conference on Telecommunications (ConTEL)*, 2003.

[103] M. Perillos and W. B. Heinzelman, "Asp: an Adaptive Energy-Efficient Polling Algorithm for Bluetooth Piconets," in *Hawaii International Conference on System Sciences (HICSS)*, 2003.

[104] X. Zhang and G. F. Riley, "Energy-Aware On-Demand Scatternet Formation and Routing for Bluetooth-based Wireless Sensor Networks," *IEEE Communications Magazine*, vol. 43, no. 7, 2005.

[105] M. Tekkalmaz, H. Sozer, and I. Korpeoglu, "Distributed construction and maintenance of bandwidth and energy efficient bluetooth scatternets," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 17, no. 9, 2006.

[106] Y. Zhou and M. Medidi, "An Energy-Aware Multi-Hop Tree Scatternet for Bluetooth Networks," in *International Conference on Communications (ICC)*, 2007.

[107] S. Lim, C. Yu, and C. R. Das, "Rcast: A Randomized Communication Scheme for Improving Energy Efficiency in MANETs," in *International Conference on Distributed Computing Systems (ICDCS)*, 2005.

[108] J. M. Kim and J. W. Jang, "AODV based Energy Efficient Routing Protocol for Maximum Lifetime in MANET," in *International Conference on Internet and Web Applications and Services/Advanced Telecommunications*, 2006.

[109] P. Sivasankar, C. Chellappan, and S. Balaji, "Performance Evaluation of Energy Efficient On-demand Routing Algorithms for MANET," in *International Conference on Industrial and Information Systems (ICIIS)*, 2008.

[110] J. Yin and X. Yang, "ELQS: An Energy-Efficient and Load-Balanced Queue Scheduling Algorithm for Mobile Ad Hoc Networks," in *International Conference on Communications and Mobile Computing (CMC)*, 2009.

[111] K. Langendoen, *Medium Access Control in Wireless Sensor Networks*. Nova Science Publishers, Inc., 2008, pp. 535–560.

[112] Y. Gu and T. He, "Data Forwarding in Exremely Low Duty-Cycle Sensor Networks with Unreliable Communication Links," in *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2007.

[113] R. R. Choudhury and N. H. Vaidya, "MAC-layer Anycasting in Ad Hoc Networks," *SIGCOMM Computer Communications Review*, vol. 34, no. 1, 2004.

[114] S. Biswas and R. Morris, "ExOR: Opportunistic Multi-hop Routing for Wireless Networks," in *SIGCOMM*, 2005.

[115] J. Kim, X. Lin, N. Shroff, and P. Sinha, "On Maximizing the Lifetime of Delay-Sensitive Wireless Sensor Networks with Anycast," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2008.

[116] Y. F. Wong, L. H. Ngoh, W. K. G. Seah, and W. C. Wong, "Dual Wakeup Design for Wireless Sensor Networks," *Computer Communications*, vol. 32, no. 1, 2009.

[117] S. Jain and S. R. Das, "Exploiting Path Diversity in the Link Layer in Wireless Ad Hoc Networks," in *IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM)*, 2005.

[118] M. Zorzi and R. R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, 2003.

[119] D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high throughput path metric for multi-hop wireless routing," in *ACM International Conference on Mobile Computing and Networking (MOBICOM)*, 2003.

[120] J. Xu, B. Peric, and B. Vojcic, "Performance of Energy-Aware and Link-Adaptive Routing Metrics for Ultra Wideband Sensor Networks," *Mobile Networks and Applications*, vol. 11, no. 4, 2006.

[121] X. Zeng, R. Bagrodia, and M. Gerla, "GloMoSim: A Library for Parallel Simulation of Large-Scale Wireless Networks," in *Workshop on Parallel and Distributed Simulation*, 1998.

[122] H. Zhang, A. Arora, Y. Choi, and M. G. Gouda, "Reliable Bursty Convergecast in Wireless Sensor Networks," in *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2005.

[123] S. J. Baek, G. de Veciana, and X. Su, "Minimizing Energy Consumption in Large-Scale Sensor Networks through Distributed Data Compression and Hierarchical Aggregation," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, 2004.

[124] Y. P. Chen, A. L. Liestman, and J. Liu, "Energy-Efficient Data Aggregation Hierarchy for Wireless Sensor Networks," in *International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine)*, 2005.

[125] S. Zou, I. Nikolaidis, and J. J. Harms, "Extending Sensor Network Lifetime via First Hop Data Aggregation," in *IEEE International Performance, Computing, and Communications Conference (IPCCC)*, 2006.

[126] M. C. Vuran, O. B. Akan, and I. F. Akyildiz, "Spatio-Temporal Correlation: Theory and Applications for Wireless Sensor Networks," *Computer Networks*, vol. 45, no. 3, 2004.

[127] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann, "Impact of Network Density on Data Aggregation in Wireless Sensor Networks," in *International Conference on Distributed Computing Systems (ICDCS)*, 2002.

[128] A. Goel and D. Estrin, "Simultaneous Optimization for Concave Costs: Single Sink Aggregation or Single Source Buy-at-Bulk," in *Annual ACM-SIAM Symposium on Discrete Algorithms*, 2004.

[129] W. Zhang and G. Cao, "Optimizing Tree Reconfiguration for Mobile Target Tracking in Sensor Networks," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2004.

[130] ——, "DCTC: Dynamic Convoy Tree-based Collaboration for Target Tracking in Sensor Networks," *Transactions on Wireless Communications*, vol. 3, no. 5, 2004.

[131] S. Pattem, B. Krishnamachari, and R. Govindan, "The Impact of Spatial Correlation on Routing with Compression in Wireless Sensor Networks," in *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2004.

[132] L. Yu and A. Ephremides, "Detection Performance and Energy Efficiency Trade-off in a Sensor Network," in *Allerton Conference on Communication, Control, and Computing*, 2003.

[133] M. Chu, H. Haussecker, and F. Zhao, "Scalable Information-Driven Sensor Querying and Routing for Ad Hoc Heterogenous Sensor Networks," *International Journal of High Performance Computing Applications (IJHPCA)*, vol. 16, no. 3, 2002.

[134] F. Zhao, J. Liu, J. Liu, L. Guibas, and J. Reich, "Collaborative Signal and Information Processing: An Information-Directed Approach," *Proceedings of the IEEE*, vol. 91, no. 8, 2003.

[135] J. Liu, F. Zhao, and D. Petrovic, "Information-Directed Routing in Ad Hoc Sensor Networks," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 23, no. 4, 2005.

[136] T. Q. S. Quek, D. Dardari, and M. Z. Win, "Energy Efficiency of Dense Wireless Sensor Networks: To Cooperate or Not to Cooperate," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 25, no. 2, 2007.

[137] L. Yu, L. Yuan, G. Qu, and A. Ephremides, "Energy-Driven Detection Scheme with Guaranteed Accuracy," in *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2006.

[138] L. Yu and A. Ephremides, "Detection Performance and Energy Efficiency of Sequential Detection in a Sensor Network," in *Hawaii International Conference on System Sciences (HICSS)*, 2006.

[139] W. P. Tay, J. N. Tsitsiklis, and M. Z. Win, "Detection in Dense Wireless Sensor Networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2007.

[140] A. Wald, "Sequential Tests of Statistical Hypotheses," *Annals of Mathematical Statistics*, vol. 16, no. 2, 1945.

[141] R. Viswanathan and P. K. Varshney, "Distributed Detection With Multiple Sensors: Part I - Fundamentals," *Proceedings of the IEEE*, vol. 85, no. 1, 1997.

[142] S. C. A. Thomopoulos, R. Viswanathan, and D. K. Bougoulias, "Optimal Distributed Decision Fusion," *Transactions on Aerospace and Electronic Systems*, vol. 25, no. 5, 1989.

[143] Y. Zou and K. Chakrabarty, "Sensor Deployment and Target Localization in Distributed Sensor Networks," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 3, no. 1, 2004.

[144] ——, "A Distributed Coverage- and Connectivity-Centric Technique for Selecting Active Nodes in Wireless Sensor Networks," *IEEE Transactions on Computers*, vol. 54, no. 8, 2005.

[145] M. Khan, G. Pandurangan, and V. S. Anil Kumar, "Distributed algorithms for constructing approximate minimum spanning trees in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 20, 2009.

[146] "Qualnet Network Simulator, Scalable Network Technologies, Inc." [Online]. Available: http://www.qualnet.com

[147] L. Kleinrock and J. Silvester, "Optimum Transmission Radii for Packet Radio Networks or Why Six is a Magic Number," in *National Telecommunications Conference*, 1978.