TECHNICAL UNIVERSITY OF CRETE
SCHOOL OF ELECTRONIC AND COMPUTER ENGINEERING
TELECOMMUNICATIONS DEPARTMENT

Interference Aware Medium Access Control

by

Nikolaos Agadakos

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE MASTER OF SCIENCE DEGREE
OF

ELECTRONIC AND COMPUTER ENGINEERING

October 7, 2015

THESIS COMMITTEE

Associate Professor Aggelos Bletsas, *Thesis Supervisor*
Associate Professor George N. Karystinos
Associate Professor Antwnios Deligiannakis

Abstract

Cross-layer protocol design for wireless sensor networking (WSN) has been in the research spotlight, due to its challenging nature and the current availability of powerful commodity radios. This work proposes TUCCOM, a cross-layer, multi-frequency channel, interference-agile and mobility-friendly, lightweight WSN protocol that utilizes various metrics found across the physical, medium access control (MAC) and network layers. Requiring no prior topology knowledge, TUCCOM can provide ad-hoc networks with self organizing and healing capability. The routing decisions are distributively made, with distance vector score-based techniques, simple interference detection and frequency channel hopping. The fundamental mechanisms are discussed and experimental evaluation results on real custom nodes are presented. TUCCOM operates on low-end, low-cost 8-bit micro-controllers, absent of any operating system. In that way, the proposed work can potentially accommodate experimental evaluations of advanced algorithms, such as network coding or backpressure routing, in interference-rich, ultra-low cost WSN testbeds.

Thesis Supervisor: Associate Professor Aggelos Bletsas

This work was supported by the European Union (European Social Fund-ESF) and Greek national funds through the Operational Program Education and Lifelong Learning of the National Strategic Reference Framework through the Research Funding Program Thales-Investing in knowledge society through the European Social Fund.

Acknowledgments

.....

Table of Contents

List of Figures

List of Abbreviations

ACK	Acknowledgment
ADC	Analogue to Digital Converter
LBT	Listen Before Talk
MAC	Medium Access Control
MCU	Micro Processor Unit
RTS	Request To Send
TUC	Technical University of Crete
WSN	Wireless Sensor Network
OSI	Open Systems Interconnection

Chapter 1

Introduction

1.1 Overview

Through the passage of time, the monitoring of specific surfaces of high interest were of prime importance for the progress of civilization. With the emergence of computers and networking it became possible to interlink distant areas, thus increasing coverage and efficiency of intelligent tracking and resource management of any location. A significant problem that swiftly arose, was the cost and size of the networks required to cover increasingly larger and harder to reach areas. Recent advances in processor and radio transceiver technology have offered low-cost, battery-powered units with relatively powerful radios. These nodes of processing and communication, have the capability of acquiring a multitude of readings through sensors, perform preliminary data processing and communicate over long ranges. This capability led to the emergence of a new type of network, the wireless sensor network (WSN), that can serve as a viable, low-cost solution in numerous problems requiring a distributed solution, or simply extensive area coverage. WSN's essential merit is its relative independence of underlying infrastructure. A number of nodes organized in an ad-hoc network can satisfy needs ranging from low demand surveillance applications, to sophisticated combat-zone mobile operations.

The usual low-cost of the nodes used for WSN's, in comparison to other solutions, provide them the ability to cover a multitude of points, offering increased spatial diversity. WSN's tend to utilize a relatively high number of nodes and be used in remote applications where central management of the network is difficult or even infeasible. The fundamental problem of reliable communication is thus extended, as WSN's are extensively resource

limited. Protocols designed for such networks have to provide reasonable power saving, while being able to route traffic towards a sink in a dynamic way, on relatively limited hardware. Thus, the issues of scalability, robustness and lifetime arise, while power versus efficiency trade-off, is a critical design principle.

1.2 Prior Art

As the field of WSN design is a mature field, a number of researchers have examined the area of cross layer design. A certain degree of OSI violation is being researched in the past few years. The main idea is to utilize metrics gathered from a lower level of OSI hierarchy, in upper levels to improve overall efficiency. The goal is to examine if this breach of abstraction can lead to more effective protocol designs. Several surveys have covered the details for the need and challenge of such an approach, as in [6], and [1] for the wireless mesh network case, which is similar in most aspects to the WSN. Cross-layer research can perhaps be mainly categorized into 3 broad categories: a) physical layer (PHY)-medium access control layer (MAC), b) PHY-routing, c) MAC-routing. While this is not definite, these 3 three combinations encompass the bulk of the effort for practical solutions. Effort has been made to extended cross design to network and transport layers; while these protocols may be difficult to implement and can require more powerful nodes, they present a venture into more holistic approach. A special case is geographical routing, where instead of node ID's and relative location, a coordinate system is assumed to be known to keep track of nodes; each node has a means of determining its location. Several protocols have been developed to cover this special case of WSN; a few will be presented as they introduce some interesting concepts to the overall cross layer design effort.

In the PHY-MAC layering scheme, the goal is to use techniques specifically designed for ad-hoc WSNs to increase capacity, spectrum utilization and energy efficiency, and import data derived from the physical aspect to the MAC aspect of the protocol to better combat node interference and energy waste. An example of this architecture is XLP protocol [9]. In XLP, a

receiver-based contention protocol was introduced. Each node sends out a ready-to-send (RTS) packet to signify to its neighbors that it has a packet to sent. Neighbors that receive this packet will then individually decide to participate in a contest for the right to propagate the generated packet. Each node checks the RSSI of the received packet, its buffer status, its remaining power and its traffic status and if all those 4 metrics are below a threshold, it will participate in the contention; otherwise it will remain dormant. This scheme allows nodes that are overloaded or have weak links with some of their neighbors, to avoid energy expenditure and reduce network congestion.

In [2], a joint MAC, routing and link layer optimization approach is proposed. This mathematical approach attempts to locate the best routing path, whether single hop or multi-hop, by solving an optimization problem under power and link quality constraints. While this scheme may produce some attractive results, it could be difficult to implement in a practical network, due to the assumptions of neighborhood knowledge each node requires.

Similar to the above, physical-routing schemes attempt to exploit physical layer knowledge to route data through nodes and channels exhibiting better overall behavior. This can lead to an increase in total network throughput, energy saving and latency. In [10], a general optimization framework for the joint link-routing problem is proposed.

In the MAC-routing combination, data gathered from the routing and MAC layers is interchanged to adjust routing patterns and node behavior in traffic intense channels and busy neighbors. An example of a light MAC-routing protocol is AIMRP [5], where node are ID-less and routing is done on a sink distance (in hops) basis. During setup each node discovers its distance from the sink by receiving a message broadcasted by the sink, incrementing its hop count by one and propagating further. Thus, each node in the network knows its distance from the sink, routing is done by transmitting to nodes that are closer to the sink. Each such node competes for the right to propagate a packet. Another example of cross layer design, intended for geographical routing, is MACRO [3]. Macro utilizes receiver based contention to propagate data toward a node with known coordinates, using the minimum possible energy. This is done by selecting to propagate packets to neighbors

with coordinates closest to the target node. In [7], a cost function solution is introduced in the geographical network approach. It considers sink relative distance, queue size of each node and remaining energy in that order of importance. While considering only a few metrics, it yielded interesting results. In [11], receiver based contention is featured for geographical routing considering link performance and energy efficiency.

1.3 Problem Statement

While the work presented above introduced some interesting concepts, it was unclear if such schemes can be harnessed to serve a network consisting of extensively limited nodes, thus the following fundamental question was the driving force behind this work. Can a lightweight, distributed communication protocol handle the requirements for communication of a WSN, in low traffic applications under unknown topologies, interference and routing path disruption? Can it be implemented and relied on as a valid solution in low-cost nodes with no prior knowledge of their surroundings? This work introduces TUCCOM, a protocol offering a hollistic approach to low-cost WSN communication problem.

Chapter 2

Medium Access Control Layer

2.1 Goals and Constraints

The basis of the design was formed upon the principles of effectiveness on limited resource MCU's and low bit rate networks. Low power availability, limited memory and sparse traffic are the essential guidelines. On this axis, CSMA is used as the foundation of the medium control. CSMA was selected as it requires no synchronization which may impose extensive overhead on low traffic networks. The mac layer of TUCCOM is a multi-channel CSMA\CA, extended with features for simple channel and link evaluation, node fairness and traffic balancing. **Initialization** Each node is assigned a channel in a static way according to its ID. This channel is known as the *native* channel of the node. Each node will then initiate TUCCOM routing discovery operation to attempt to establish a functional network. The routing procedure is presented in chapter 3. Every node wishing to contact any other node of the network, with no extra information, will always transit to its target's native channel at first. Any node hopping away from each native channel, due to some reason, is known as an immigrant node.

2.2 Physical Layer Aspect

In order to avoid traffic ladden or simply interference-heavy channels, a simple channel evaluation mechanism was developed. A node gathers rough data about each visited channel and its own native channel. This gathering is executed in the following way. Each time a node visits a frequency channel (herein after referred to as channel) to attempt to contact another member of the network, it first listens to the medium as per the Listen-Before-Talk

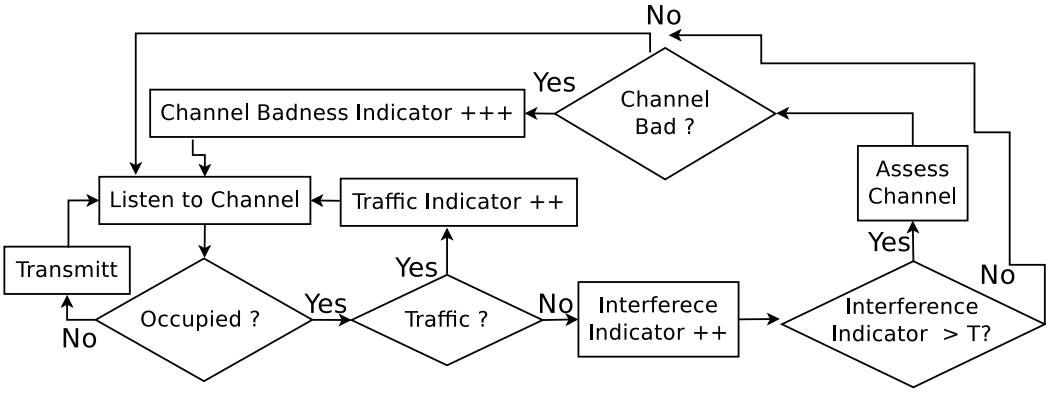


Figure 2.1: The scheme used for channel evaluation.

(LBT) procedure. If the listening node detects elevated traffic activity, it will register that information by increasing a variable called *Traffic Indicator*; if the channel detects increased noise levels, it increases the *Channel Badness Indicator* of the channel. The node will use these indicators to make adjustments to its timeout clocks, retransmissions chances when collisions occur, sleep periods and which channels to avoid when attempting to reach to another node. This data registering can help decrease energy expenditure in congested or jammed channels. The block diagram of the channel evaluation mechanism is presented in figure 2.1. The **Traffic Indicator** (TI), is a variable associated to each channel individually. Each time a node attempts detects foreign traffic when visiting a channel it increases its traffic indicator, in the following way. The first time a foreign message is overhead the TI is increased and a 2-bit flag indicating consecutive times a channel was found occupied before a successful transmission occurred is set. Ti increases linearly for low flag values and exponentially for high flag values. This aggressive approach was chosen as low traffic networks where the design target. With this scheme, nodes visiting high traffic channels can adjust their behaviour for smoother network behavior. Transmitting nodes will only transmit with a probability p, decreasing as the channel TI increases. A node that has lost contention for too many many rounds will issue a suppress command on its messages, forcing overhearing nodes into extended slumber, to minimize

competition. The standard exponential backoff schemes are also invoked.

Similarly to the TI, the Channel Badness Indicator (CBI) is used to modify node and channel interaction. When a node enter the LBT procedure, unexpectedly high channel occupation is deemed as suspicious behavior. The node will then increase the CBI of the particular channel, staying vigilant for future interference. When CBI passes a threshold, the channel is branded as unusable and nodes will avoid the channel in future communication. Nodes native to that channel or simply residing, will relocate to a new channel according to the selected channel hop mechanism. This channel hop mechanism can be a simple linear hop. Immigrant nodes will always attach their new status to their messages, so new neighbors can access them directly to their new channel without wasting energy. The thresholds for the above indicators were heuristicly determined.

2.2.1 Channel Allocation and hopping

TUCCOM operates on a set of orthogonal frequency channels which are divided into 3 categories: 1) sink channels, 2) node channels and 3) burst-access channels. **Sink channels** are channels assigned to sinks and are also used during network discovery by all nodes. Any node that has direct sink access or wishes to check if a sink happens to be within range, must change to a sink channel and transmit. Given the fact that it can be assumed that multiple sinks will be beyond each other's radio range, a single channel dedicated as sink channel , will suffice. Sink channels can be used as backup channels for nodes, if all other channels become corrupted.

Node Channels are those channels that are allocated for node-to-node traffic. Each node resides in one, performing any programmed operations and waiting for incoming traffic. Each node has a **native channel**. The native channel is the default channel of a node. Any other member of the network that attempts to contact a specific node and has no further info about its whereabouts, will always attempt to make first contact at its recipient native channel. The native channel is fixed for each node. To maintain network

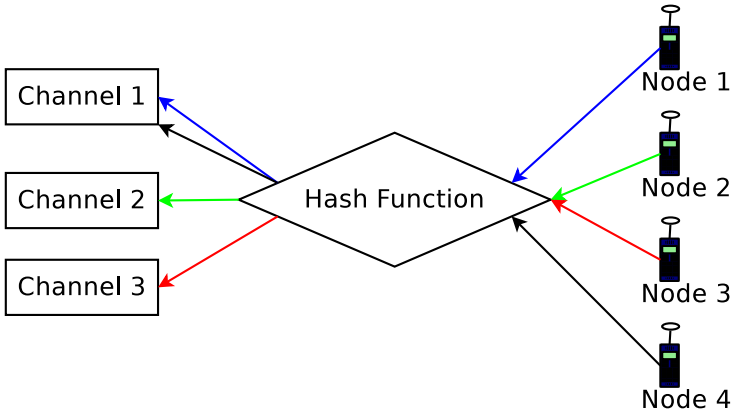


Figure 2.2: Static channel allocation as used in TUCCOM.

functionality, a node whose native channel becomes congested can migrate to another channel. An immigrant node will always attempt to return to its native channel, to restore network equilibrium. An immigrant node will poll its native channel to check its state, in terms of noise floor. The frequency of this poling decreases exponentially with the number of times the native channel is found corrupted. The current channel a node occupied is called the residing channel. A native channel is also a residing channel.

Burst access channels are dedicated to burst-type traffic. Any node wishing to transmit multiple packets back-to-back to a recipient, will utilize this channel. The transmitter jumps to the residing node's channel and attempts contact. On a success, it transmits the first packet of the burst exchange, and notifies the recipient to jump to a burst access channel (of a transmitter choice). Then the exchange is carried out normally, with a series of packets and acknowledgments. These type of channels are not mandatory for normal operation and can be omitted if there are not enough available channels to use. **Channel Allocation** can be done through any algorithm providing a single channel that can be used as the native channel of a node, for each node of the network. In this implementation a simple cyclic hash function using a node's ID was chosen, seen in Fig. 2.2. This simple approach, is used for initial channel allocation. Nodes can assess their channels during their operation and reallocate if necessary.

The total number of channels is a matter of balancing resource ex-

penditure and traffic spread over frequency. High number of available channels provides the network with versatility, interference resistance and traffic spread over the available frequencies. This approach has merit in dense, high traffic networks. A low number of channels approach provides interference resistance while, reducing the overhead required to locate nodes across the channels. This approach is better suited to sparse, low traffic and energy limited networks.

2.3 Link and Medium Access Layer Aspect

2.3.1 Link level transmission

When a node wishes to transmit a packet to one of its neighbors, it switches to that neighbors native channel and sends out a message to that node. If the recipient is currently on its native channel, it will answer (if ACK is requested) under a normal data-ack scheme. If the sender fails to reach its recipient, it will attempt to recontact at a later time, provided the number of these retransmit attempts does not exceed the maximum number of retransmissions allowed in the network, which is a preset variable. On the occasion that the target channel is deemed as unusable, the sender will switch to the next channel assigned to its recipient according to the channel allocation function used (in this implementation a simple linear function was used).

2.3.2 Medium Access Control

Access to the medium is regulated via an extended CSMA\CA scheme. An overview of its mechanics and a presentation of these extensions is discussed. The block diagram of the overall mechanism is presented in Fig. 2.3.

A node attempting to transmit to another node, will first use the algorithms described in 2.2, 2.3.1, to locate the channel of its recipient. It will then follow the following steps.

1) Initiate LBT procedure. Before each transmission, the node will listen to

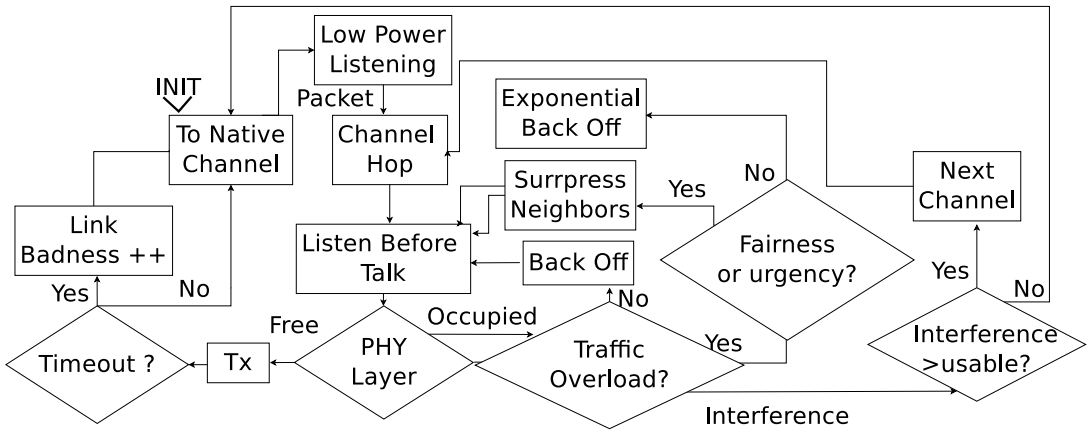


Figure 2.3: Block diagram of TUCCOM's MAC layer.

the medium for a predetermined interval. If the medium is free of traffic it will transmit a message to its recipient along with the option for acknowledgement (in this implementation ACK is always on).

2) If the recipient receives the message, it will acknowledge reception. Should a timeout occur, the sender will attempt to resent its message, provided the total attempts do not exceed the maximum retransmissions. Exchange is complete at this point.

3) If the channel is found occupied due to foreign traffic, the sender will backoff and sleep. The backoff scheme is a standard probabilistic exponential backoff algorithm. The more a channel is found occupied the more reserved the node's approach. When a channel is first found occupied, it will back off for 2 message length time intervals and attempt to resent. The node will continually double this waiting interval up to 8 times the message interval. Should the channel be found occupied with a foreign packet again, probabilistic wake up is introduced. The node will attempt to sent with a probability $p = 0.9\% - 10\%$per channel occupation. It is unlikely that so many consecutive overhearing will occur in a low traffic network. Should a message be labeled as **urgent**, a node will issue a special surpression command, so that overhearing nodes will sleep for an extended period of time (5-10 message intervals), trading off network latency for urgent message expedious delivery.

A node contending in a "bad" channel (one with a high traffic indicator), will invoke the cautious back off mechanisms upon hopping to minimize lost energy.

Immigrant nodes and Channel Search

Any node that has moved from its native channel is called an immigrant node. A node that has migrated will always append its status on any messages it sends. In that way overhearing nodes will not waste energy by attempting to contact it in its native channel.

When a node wishes to contact a neighbor that has left its native channel, it will transit to the next channel for that node and attempt to locate it there. A node will only perform this once, unless that particular neighbor is the only remaining neighbor, thus the only relay of information. The hopping mechanism utilized in this implementation is a hash function seeded with the node's ID. Every node has access to the same function and can calculate the channel-hop sequence of every member of the network.

2.3.3 Frequency Allocation

A fundamental part of any multi-channel scheme, is channel allocation. While a great number of algorithms exist that solve this problem with adequate efficiency under a centralized approach, they may be unfeasible under a distributed approach, for a low-cost sensor network. TUCCOM's approach is a statically receiver based, preallocated approach. Each node is assigned to a channel based on node ID. This initially allocated channel to a node, is it's **native channel**. The sinks are assigned to the sink channel(s) and burst channels are available for all nodes and sinks.

2.3.4 Channel Assessment

Nodes can assess channels to detect interference. Each node sets a time frame where it estimates the noise floor of the target channel. The threshold is relative to the average neighbor signal RSSI the node has encountered

and radio sensitivity. For example if the average neighbor RSSI is -70 dBm and radio min sensitivity is -80 dBm, the threshold for interference detection should be set at -80 dBm, at most. During that time interval a node has set, if the channel is occupied for the majority of the time, with no significant periods of calmness, a metric call Channel Badness is increased. When channel badness surpasses a preset programmable threshold T, the channel is labeled as “corrupt”. Node residing at corrupt channels hop to the next uncorrupt available channel. Channel badness is decreased over time and increased each time a channel is polled and found with high noise levels.

Chapter 3

Routing Layer

The routing layer was designed with the fluid nature of WSN communication in consideration; abrupt node malfunction, phusical channel alterations along with scarce bursts of traffic. On this basis, TUCCOM was designed to provide a means of sink-oriented, hop-centric communication, where each node utilizes available information for its immediate vicinity to select the optimal next hop recipient. Nodes do not require prior knowledge of the network and no supporting infrastructure is needed except a number of nodes serving as sinks. TUCCOM can support a multitude of sinks.

The main idea behind the routing algorithm is the notion of **Routing Value (RV)**. RV is a score that describes how valuable a node is, as a gateway and an overall member of the network. Nodes with high RV scores are the preferred relays of information. Nodes always propagate their info towards neighbors with the highest RV. RV is calculated dynamically and locally, at each node, for every neighbor the node has discovered in the network. It is measured by adding together scores derived from a neighbor's remaining power, RSSI level, traffic status, neighbor behavior, the number of reliable neighbors a neighbor reports it has, and most importantly sink access and sink RSSI. The above are known as the **basic commodities**. RV is recalculated each time the node detects a basic commodity change, and different commodities have different impact; the most important ones being Sink Access, Remaining Power, and Direct Link Access. The impact of each commodity is expressed by a number, known as the **Commodity Value (CV)**. In essence, it is a quantification, into an integer score, of important network characteristics and a categorization of their importance in the routing procedure.

This layer works in conjunction with the MAC layer described before, in that

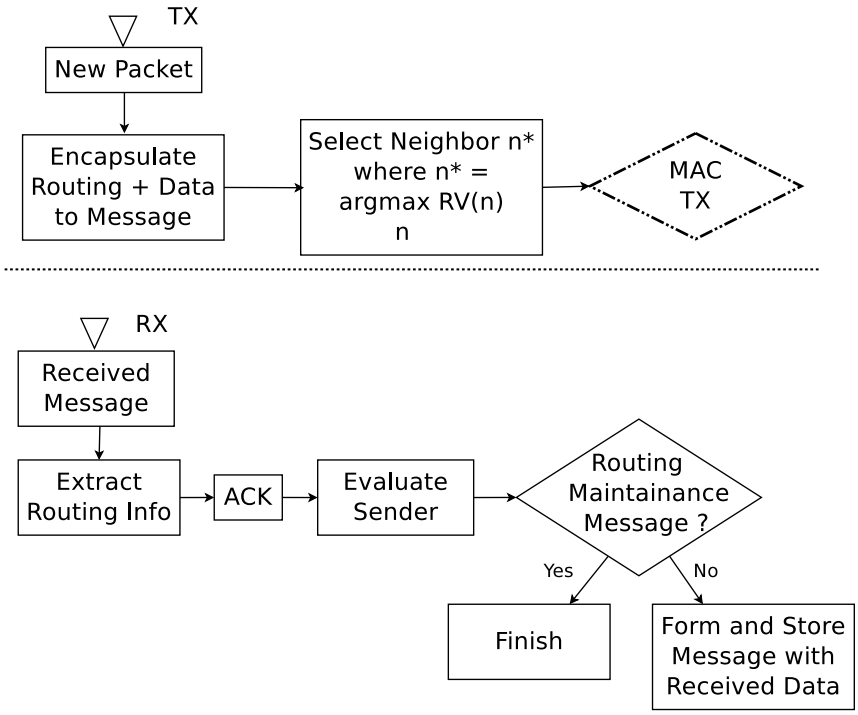


Figure 3.1: Routing layer block diagramm.

it utilizes gathered data for traffic, noise level for each channel and data for the availability of discovered neighbors. The routing layer uses this gathered data as input for its decision making. Congested channels are avoided and unavailable neighbors have their routing values reduced. Data across the various layers are encapsulated in sent messages, whose structure is presented in Fig. 3.2.

This greedy approach aims to combat several parameters that cannot be anticipated in WSN communication, i.e a temporary communication link failure. Each node individually calculates the next hop recipient, for each packet. Previously selected relays might fall out of favor as newly discovered neighbors emerge or old ones become more reliable. An overview can be seen in Fig. 3.1.

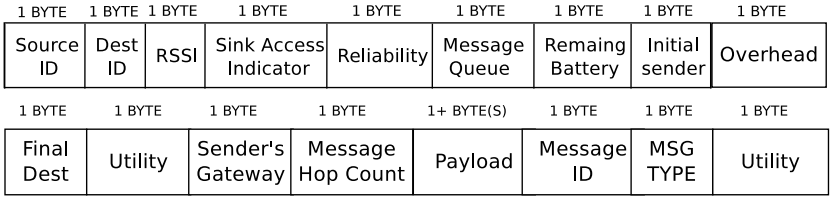


Figure 3.2: TUCCOM message stracture.

3.1 Message Structure

Before explaining the specifics of RV calculation and routing decision making, the message structure of TUCCOM is presented. The message used in TUCCOM is a 17 byte long packet, encompasing all information across various layers. The message length must be as short as possible, to conserve energy while being able to convey the necessary information. An overview of the various fields is presented below, an overview is presented in Fig. 3.2.

Source Id: The node ID of the last hop sender.

Destination Id: The node ID of the next hop recipient.

RSSI: The RSSI of the current link, as seen from the sender.

Sink Access Indicator: A variable indicating the quality and distance this node (the sender) has from a sink. It's value are a) 1-hop access, b) 2-hop, c) 3-hop, d) 4-hop, e) 5-hop+, f) no access.

Reliability: This field is used to denote the number of neighbors that also possess Good sink access (direct, or 2-hop), for extra versatility. This field can be also used to include an estimation of the overall behavior of this node in the network. There are 3 values, a) good, b) average, c) bad. This value is used to adjust this node's routing value according to its behavior. It is possible that a node with otherwise perfect resources, to behave poorly. This field aims to address this issue.

Message Queue: The number of messages this node has waiting to be transmitted and an evaluation of passing traffic through this node.

Remaining Battery: A rough estimation of the remaining battery. This estimation is derived from the expected energy expenditure of each reception and transmission, and the number of wake up cycles. It is used to re-route

traffic to less active nodes if possible.

Initial Sender: The node that generated the value in the payload field, originally.

Overhead: The number of messages used to maintain routing paths and network connectivity. This can be used to diagnose problematic nodes in the network.

Final Destination: The final destination of the original message.

Utility: This field serves for the utilitarian features of TUCCOM and for further extensions. Neighbor suppresion, burst access request, node immigrant status, sink immigrant status (if a sink has migrated to a different channel), are all activated by this field.

Sender's Gateway: The chosen gateway of the sender. This is used to avoid routing loops. For example, if the receiving node has the same gateway as the sender, then the routing value of the sender for the receiver, is -1 (has no value as a relay).

Message Hop Count: How many hops the original message has performed.

Payload: The data meant for the final destination.

Message ID: This field, in conjunction with the Initial Sender ID, is used to uniquely identify the message. It is also used as a rough timestamp. Nodes will compare the message ID's between messages from the same node to determine if their data are up to date or obsolete. For example, in node A informs node B that node C has hopped to a new channel, and that update has a message

Message Type: The declare type of this message. It can be a) Data message, b) ACK message, c) Initial Discovery, d) Routing maintenance message or e) Routing Discovery message (node has to rediscover the network).

Routing Tables Structure

An important aspect of TUCCOM, is the organization of kept routing tables. The tables used must hold all necessary information with out growing exceedingly large, thus an attempt has been made to store only critical information

for each neighbor. The layout for each table entry along with example value corresponding to the network presented in 3.3, is displayed in 3.2.1.

The basic metric attributes of TUCCOM are viewed as commodities in the sense that they are resources that can be acquired and utilized and are of different value to different parts of the network. For example, a high direct link RSSI, is of higher importance to a node whose neighbors all have weak signals, than for a node whose neighbors' have high RSSI. The Commodities are listed and defined along with their qualitative and quantitative (in parenthesis) value set. The values listed are the final CV for each one; the weight with which they influence the overall RV (Tab. 3.1) is calculated.

1) Sink Access: is the most important commodity. It stands for estimated distance (in number of hops), between this node and the sink. It is measured in hops.

Values: 1 Hop (61), 2 Hop (36), 3 Hop (21), 4 Hop (11), No Access (-100).

3) Remaining Power: is the second most important commodity. It is the self reported power level of a neighbor.

Values: High (15), Medium (5), Low (0), Very Low (-15), Critical (-25).

2) Direct Link RSSI: is the third most important commodity. It is the latest RSSI estimate of the evaluated neighbor. The CV for each Commodity except the last 2 is calculated by comparing the RSSI of the latest communication with the overall average RSSI. A neighbor is awarded +\-1 RV per dBm above or below the overall average RSSI the node has calculated, up to +\-10, respectively. The last 2 values penalize a neighbor for having a signal close to the radio's sensitivity.

Values: Excellent, Good, Average, Weak, Poor (-15), Unstable (-30).

4) Sink RSSI: is the estimated RSSI between an evaluated neighbor and a sink, if that neighbor has direct sink access.

Values: As in Direct Link.

5) Traffic Status: is the estimated traffic level of a node. Nodes attempt to avoid high traffic neighbors, in an effort for load balancing. This is done by slightly reducing a neighbor's RV.

Values: Low (0), Average (-5), High (-15), Overloaded (-25).

6) Neighbor Behavior: is the overall behavior of a neighbor as a relay

of traffic. While a neighbor's self reported commodities mighty be of high value and thus appealing, it might not be a good relay choice. This can be due to node malfunction, physical partitioning or temporary channel fade. Every time a neighbor is chosen as a relay and cannot be reached (ACK loss or total contact failure), a metric called "Node Badness" is increased, reducing its neighbor behavior commodity and thus, its overall RV. A variety of factors can contribute to this fact, such as faulty hardware, fade bursts, physical partitioning etc.
Values: Good (0), Average (-5), Acceptable (-15), Poor (-25), Very Poor (-40), Bad (-100).

7) Neighbor's Reliable Neighbors: is a node's self-induced report about its neighborhood reliability. This is essentially an averaging of the routing value and behavior of all registered node's neighbors.
Values: A : at least one neighbor with direct Sink Access. (9) B : at least one neighbor with 2 hop Sink access. (5) C : one neighbor with 2 hop Sink Access. (1) D : All neighbors have Sink access at 3 hops, or more (-1). E : No neighbors or none with any Sink access (-4).

3.1.1 Commodity Value Assignment

The most important element of the proposed routing methodology, is the sorting of importance of the various commodities. It is imperative to assign the highest impact to the most important network characteristic. For example, since most communication is sink-oriented, the most valuable resource could be sink access, measured in hops. Nodes closer to the sink are invaluable as relays, as they are able to forward any traffic towards its ultimate goal. In that sense, an otherwise average-scored neighbor with direct or 2-hop sink access, obtains a high RV. Commodities' value range varies according to their contribution as presented in Tab. 3.1. In general, any networking feature that helps promote network functionality is usually positive, while any feature that deviates from expected operation, such as timeouts, low battery, high traffic or low RSSI, is negative.
Limit values of some commodities are special cases requiring a different ap-

Commodity	Values	Weight
Sink Access	Positive	61%
Remaining Power	Any	15%
Link RSSI	Any	10%
Sink RSSI	Any	5%
Reliable Neighbors	Any	9%
Traffic Status	Non-positive	0-20%
Node Behavior	Non-positive	0-100%

Table 3.1: Commodity impact and value range.

proach. For example, while every value of Sink Access provides positive RV, having no sink access at all negates all RV from other attributes, as a node that does not have any neighbor that has discovered a sink, is useless as a relay. To reflect this, a node with no Sink Access is assigned a negative RV. Similarly the last two values of Remaining Battery impact negatively (reduce the CV), to encourage nodes to seek alternate paths, in an effort to keep nodes from wearing out their batteries. Finally, the **RSSI** assessment and value assignment is performed by comparing the link's RSSI with the overall average RSSI the node has from all discovered neighbors and then compared to the RSSI specified for its radio hardware. For example, for each dBm the RSSI is found above or below the average link RSSI the node scores a +\- 1% CV respectively. However, if the RSSI is found to be near the operational threshold of the radio the nodes are equipped with, that neighbor is penalized heavily as communication failure probability increases near that breaking point.

3.2 Routing Value

The overall RV of a neighbor is calculated by the summation over all its CV's. Rv is registered for each discovered neighbor.

$$RV(n) = \sum_{i=1}^{|C|} CV_i(n) \tag{3.1}$$

Where: CV_i s the value of commodity i, as viewed from current node and

i $\in \{C\}$, where C is the set of available Commodities.

3.2.1 Routing Decision Example

Each time a node wishes to transmit a packet, it checks the discovered neighbors' RV's and selects the one with the highest value as the next-hop recipient. Due to its consideration of a wide variety of important metrics, RV can by itself serve as a valid routing indicator. Each node decides locally and dynamically which of its neighbors will be the gateway for each message it has to transmit. Nodes with negative RV are never selected as gateways. All nodes maintain lightweight routing tables to store the information required. To better illustrate the mechanism, an example network is presented in Fig. 3.3. Consider the table presented in Fig. 3.2.1, to be the current routing table of node 3. Node 3 has finished network discovery and has registered 3 neighbors: nodes 1, 4 and 5. These neighbors are evaluated and their final RV is registered. Whenever node 3 wishes to transmit a message, it will select the neighbor with the highest RV. In this case this corresponds to node 1. This is indeed a logical decision, as node 1 has direct access to a sink. Referring again to Fig. 3.2.1, node 1 has inferior RSSI, Power and Traffic CV to node 4; it is the gravitas of the Sink Access commodity that significantly raises its RV. Node 4 is also an attractive choice for a gateway as it has 2-hop sink access and its other Commodities are of high value. In contrast, node 5 reports node 1 as a gateway, thus it is automatically of no use to node 1 as a gateway; to signify this, its RV is set to a negative value.

To signify the varying importance of each of the above factors, each one has a different impact on the overall Routing Value calculation. One way to do so, also the one currently used, is to let each commodity have different value ranges, as mentioned above. This enables the most important commodities to be responsible for a greater percentage of the final routing value and also have much more accuracy reflected. For example, enabling Sink Access to be responsible for 51% of the overall routing value, allows this metric alone to imply that if a node has direct sink access, however weak, always

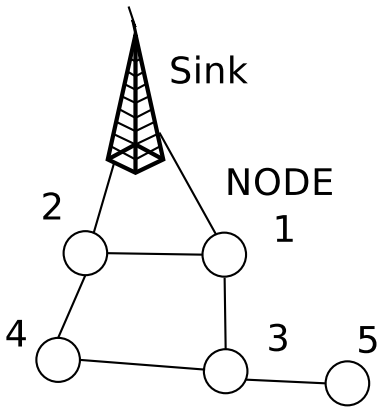


Figure 3.3: An example instance of an established network.

Node Id	1	4	5
Routing Value	67	50	-100
Sink Access	Direct	2-hop	None
Direct Link RSSI	-45 dBm	-35 dBm	-51 dBm
Remaining Power	Average	High	High
Sink Link RSSI	-60 dBm	None	None
Traffic Status	High	Low	Low
Neighbor Status	Good	Good	Good
Neighbor's Reliable Neighbors	1	1	1
Gateway	Sink	2	3
Last Message Id	34	-	28
Last Message Source	4	-	5

Figure 3.4: TUCCOM's routing table structure overview, with node 3's view of the network set as an example.

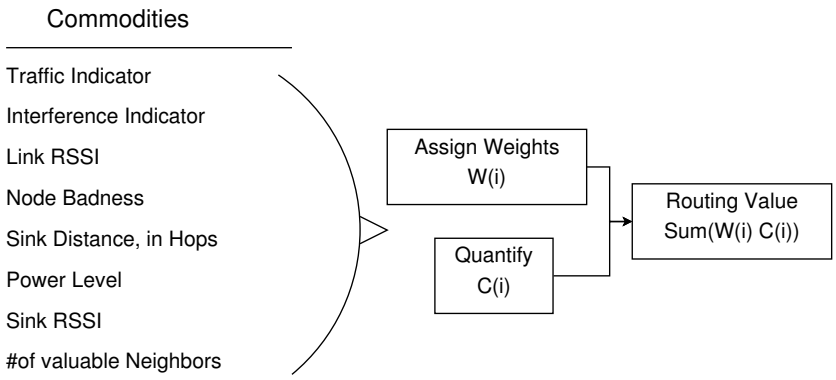


Figure 3.5: Routing value calculation overview.

has some value as a routing relay for the network.

3.2.2 Routing loop avoidance

In any non centralized network, closed loops among the routing paths can occur. These "black holes" waste network energy and disrupt the flow of information with potentially catastrophic results, as critical data can be lost. To combat this problem, steps have to be taken to ensure nodes can identify such a loop and rearrange their routing pathways. In TUCCOM lightweight routing tables, message identification and Time-To-Live (TTL) encapsulation is utilized.

Routing tables are used to verify the last hop sender of any received message. If a node receives a message from its gateway, it means that this gateway is compromised and can no longer route its messages towards the sink. The node will choose the next neighbor with the highest RV that it has stored in its routing tables, as its gateway. It will also nullify (drop it to -1) the routing value of any node that cites the compromised relay as its gateway and reports no other neighbors with sink access. Any neighbor, that cites as a gateway, a node that is known to be comprised, has little value; routing information through those nodes will only waste extra energy and potentially form unnecessary long routing paths or routing loops. In addition, a node will never forward a message to its last hop sender or to its original source.

Message Identification is performed by jointly examining a message's ID

and the ID of its original source. Each node keeps a record of the last received message from each neighbor and the last 5 messages the node itself propagated. Should a message with a recurring message ID-source combination be received, it will not be forwarded. It should be noted that each time a new message is generated, it is assigned a new message ID. A node that receives one of its own messages brands its gateway with no RV and selects as a gateway the next registered neighbor that does not cite the old compromised node as its chosen gateway.

Time-To-Live is also used, to reduce the occurance of routing loops. A node that receives a message that has been forwarded more times than its predetermined TTL count will not be propagated. TTL is determined before network establishment and currently is a hard coded variable (valued at 5 hops).

3.2.3 Partition Recovery

An inherent problem in WSNs, is network partitioning. This can occur from a myriad of reasons such as extensive channel fading, physical blocking, or node failure. TUCCOM is equipped with a mechanism to maintain functionality despite such mishaps.

As nodes are power limited, the main idea is to find alternate routes while being subjected to power conservation constraints. Great care must be taken to avoid exhausting a node's battery with intense network rediscovery attempts. Rediscovery attempts become increasingly sparse, as the node fails to reach a node with positive routing value. This is based on the concept of temporal locality. Channel fading may severely affect communication for a short amount of time, but physical partitioning tends to be lengthier. As such, if a node cannot reach some reliable neighbor within a short interval, it tends to wait out increasingly longer periods to conserve its battery.

A node will transmit a short packet sequence called Complete Routing Info Request (CRIR) packet. This consists of all the partitioned node's routing info and an indicator requesting an answer from each recipient node. Nodes with negative RV are kept in memory but are not considered of any value as

relays; a node will keep broadcasting CRIR's until a neighbor with positive RV is reached.
A node will also attempt to contact neighbors that cite it as their gateway, if it has not received any packet from them, for too long. This is done in an attempt to counter hidden terminal type interference, which is only visible to the node's neighbors but not to itself.

3.2.4 Mobility Support

TUCCOM can be tuned to support limited mobility. By increasing the negative impact of the Node Badness on the Node Behavior commodity, nodes render neighbor's obsolete as gateways faster. Thus, mobile nodes avoid wasting energy by attempting to contact out of range nodes. An example for this is presented in Sec. 4, where a single timeout event nullified a gateway's RV and caused the moving node to look for alternate relays.

3.3 A qualitative example

A simple example of operation is presented. Suppose a network of 7 nodes and a sink, all equipped with similar hardware. Assume half duplex communication and a single radio per node, capable of switching to multiple frequencies. All nodes have a similar start; maximum power level, no overview of the network and no sink access. The operation is separated into time frames for explanatory reasons; TUCCOM is asynchronous. Suppose a topology and radio connectivity as shown in Fig. 3.6. Each node in the network attempts to find a neighbor with positive routing value, or direct sink access. To achieve this, each node sends out short broadcast packets requesting full routing data on all available frequencies, starting from those frequencies that are preallocated to sinks.

During **frame 1**, shown at Fig. 3.7, Nodes A and G manage to establish links with the network sink. The rest of the nodes, having no feedback from the new information of nodes A and G, retain their network discovery efforts

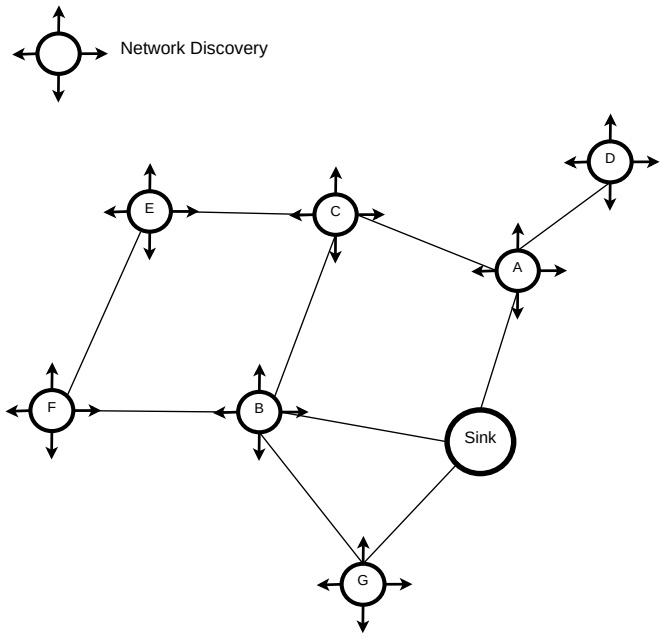


Figure 3.6: Initial Snapshot. Edges represent radio connectivity.

to establish connections of positive routing value.

During **frame 2**, shown at Fig. 3.8, Node B manages to contact A, receiving its updated information. Node A has Direct Sink Access, over a strong link, maximum power and is thus a high value node for routing information. Node B updates its own routing information; it will now notify any node contacting it that it has a very valuable neighbor that can relay information. It is important for B to also transmit the ID of its relay to avoid routing loops. As such along with B's own routing information, which is remaining power, reliable neighbors, estimated traffic and sink access (in hops), it will also transmit that its preferred gateway is node A. Node B will transmit that node A has direct sink access, and will also transmit the RSSI that A self-reported it has with the sink. With this information, any node contacting B will be updated about its routing info and which is preferred gateway. For this example, it is assumed that node D failed to contact A during this frame; it will continue to broadcast Complete Routing Info Request messages (CRIRs).

During **frame 3**, shown at 3.9, Nodes C and F establish a link with node

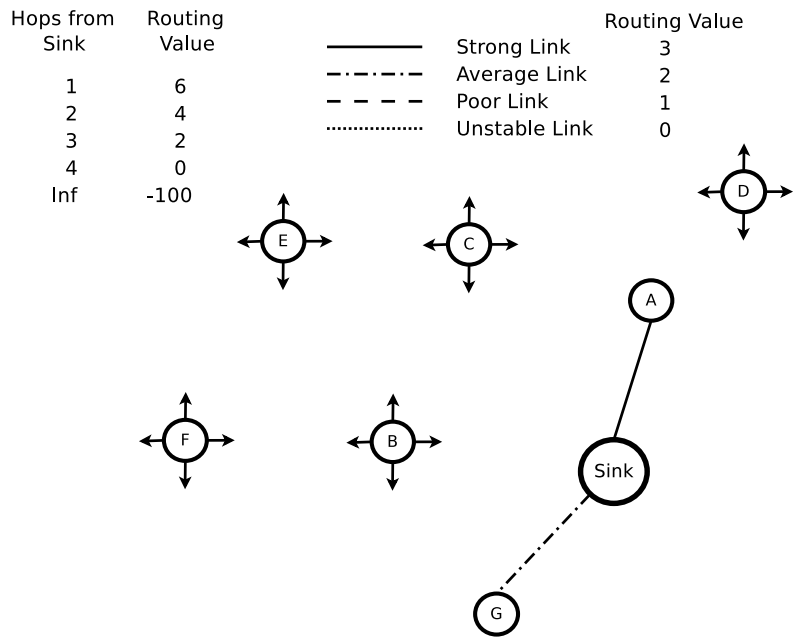


Figure 3.7: Network at frame 1. Node A and G have established links or positive Routing Value.

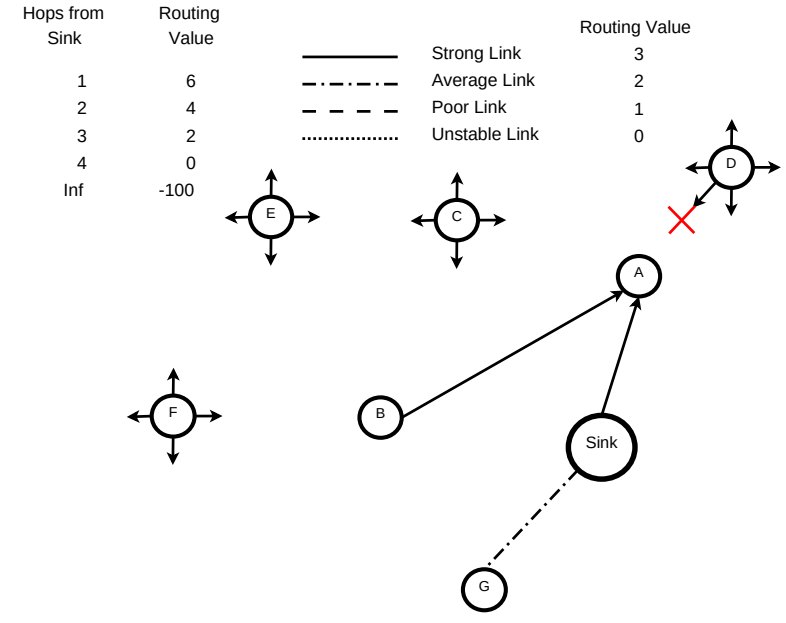


Figure 3.8: Network at frame 2. Node B establishes link with A. D's attempt fails, thus D continues to broadcast beacons.

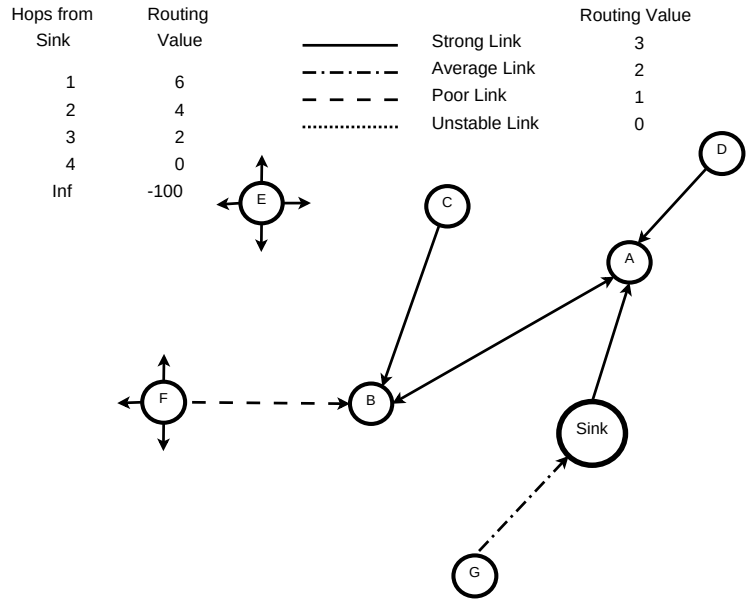


Figure 3.9: Network at frame 3. Nodes C and F establish link with B.

B. Node B during their exchange sends them its updated routing info. Node C assesses that since no other nodes exist with higher routing value than B, B is sound choice for relaying as it has a two hop sink access over, what B reports, strong links. As such node C decides to end its network discovery, settling for b as a routing relay. At a preset low-frequency time intervals, Node C will attempt to find new potential neighbors in an attempt to keep a valid overview of its neighborhood. These attempts become increasingly sparse as no new nodes are discovered. Node F on the other hand, while it has managed to contact a neighbor with sink access, it has done so over a poor link. Given the fact that node F has no other valuable neighbors and its only gateway is on a precarious link, it will keep broadcasting messages to discover other neighbors at an constantly reduced frequency.

During **frame 4**, shown at Fig. 3.10, Node E contacts node C and F during its network discovery broadcasting sweep. Nodes C and F both have similarly valued commodities. Both have 3 hop sink access, the same gateway, maximum power, just one reliable neighbor (B) and the same traffic level. The only difference between the C and F from E's point of view, is the quality of the path from E to its neighbor's gateway, A. With otherwise similar

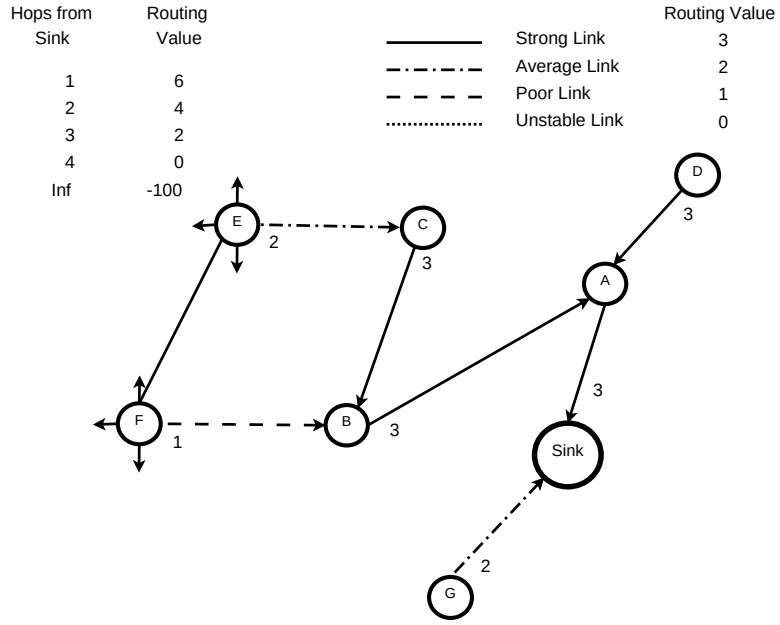


Figure 3.10: Network at frame 4. Node 3 establishes links of positive routing value with C and F, and joins the network.

commodity values, E will decide with node to entrust as a gateway based on the comparison of the paths $E \to C \to B, E \to F \to B$. Assuming a value of 3 for a strong link, 2 for an average and 1 for poor, path $E \to C \to B$ has a score 5 ,while path E $\to F \to B$ a score of 4. As such, Node E will select C as its preferred gateway. The value for the paths is subject to frequent change due to the nature of the wireless medium. Each time a node contacts another node it recalculates its routing value with any new information.

Chapter 4

Implementation

TUCCOM was implemented on custom nodes featuring an C8051f320 MCU [8] and a Chipcon CC2500 Texas Instruments proprietary radio [4]. The node can be interfaced via an extension board to a multitude of peripherals and is powered by a pair of standard AA batteries. The radio supports Low-Power-Listening (LPL), stated as Wake-On-Radio (WOR). This functionality enables the node to enter low-power sleep mode, periodically switching on its radio to probe the channel for incoming traffic while maintaining the MCU and any other digital peripheral into sleep mode. On successful detection of legitimate incoming traffic, the radio receives the message, signaling the MCU of the newly arrived traffic, therefore waking it upon radio reception. This functionality enables the node to operate for extended periods of time and indirectly forms a frame-like architecture, in which each node exhibits its own wake up and reception pattern. This behavior can reduce wasted energy due to traffic not meant for a node. A sample instance of WOR operation is shown in Fig. 4.2. The node along with its extension board is presented in Fig. 4.1.

4.1 Software

TUCCOM was implemented on C8051 architecture MCU's, written in C. The available resources of the MCU were 16 KB of Code RAM, 256 runtime Data RAM and an extra total of 1 KB external RAM used for variable and structs.

Figure 4.1: An icube, the node used in the implementation, shown in ip64 casing.

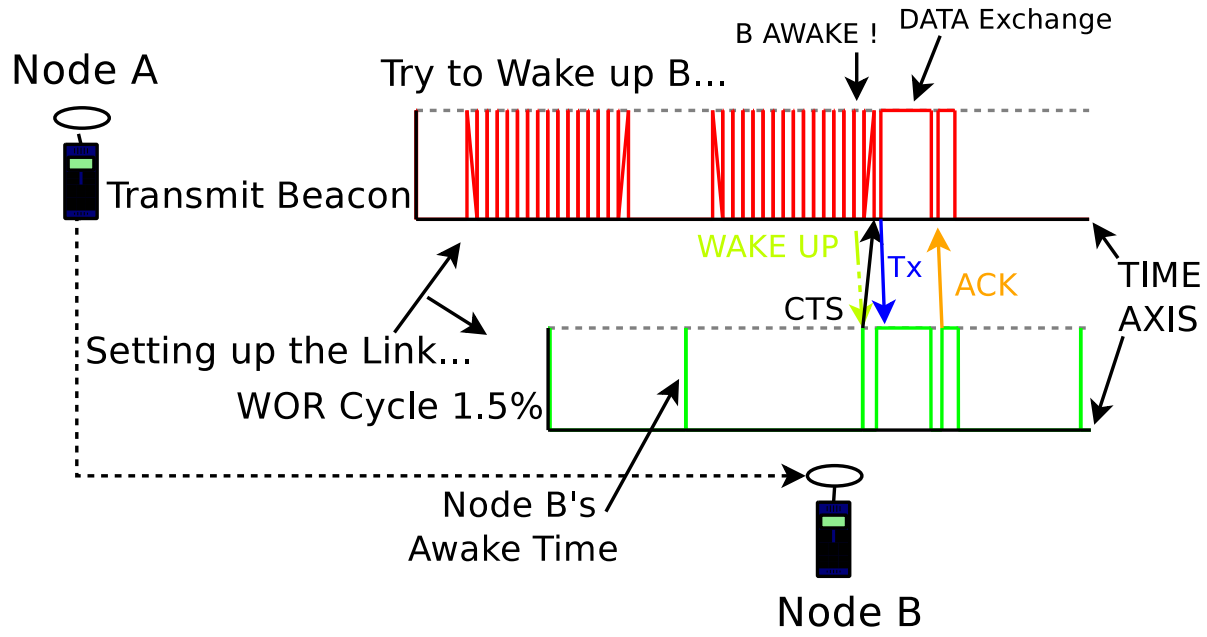


Figure 4.2: Handshaking, data-transfer and acknowledgement in the WOR scheme.

4.1.1 Programming and Synchronization Considerations

One of the fundamental challenges of the implementation process, was the transmitter-receiver-MCU synchronization procedure. The principles of structured and logical programming require the delegation of functionality to short, well defined and concise functions; the need for this is augmented by the reliance on the MCU's code overlayer. The code overlayer is a complex program that allows the compiler to build a precise function call tree. This enables the MCU to have an accurate image of exactly what resources each function needs and which function must run simultaneously. With this information the MCU can allocate its limited resources much more effectively, by assigning memory space only to the function that have a probability to run simultaneously; the resources of the rest are released. The issue with this otherwise correct practice, is the delay it introduces into the function call procedure of the MCU's scheduler. To illustrate this consider the following

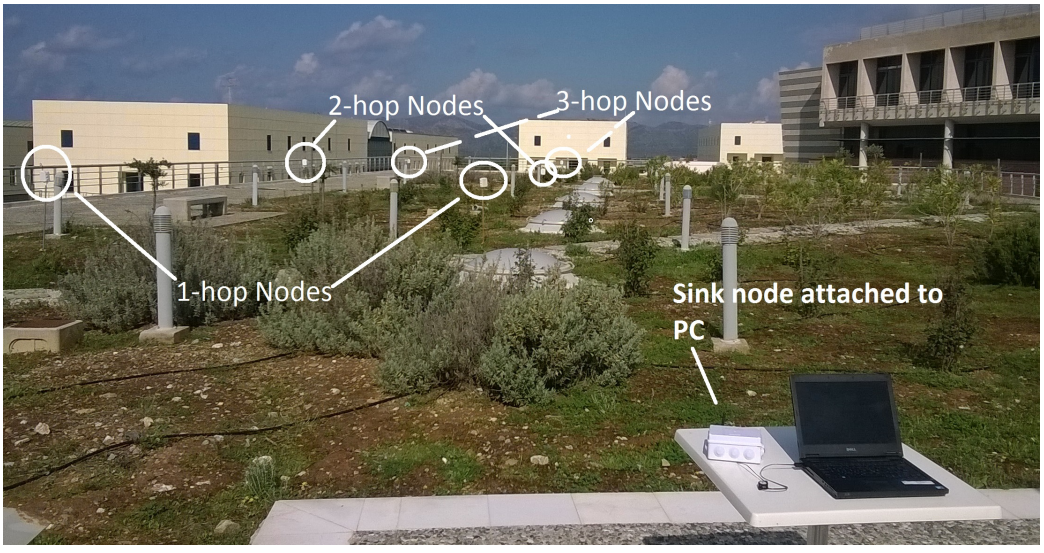
example. Suppose node A is receiving a message from node B, both with identical code and nominal capabilities. During the chain of function calling, each node despite similarities, exhibit a different response time and delay between function calls. This is due to the different RAM state each node displays at any given time (different volume of data to be stored or loaded), slight dissimilarities in the clocking system and due to slight hardware deterioration. This drift in response time can lead to loss of synchronization at any point of the exchange procedure, resulting in loss of communication. This seemingly minor problem, can inflict serious disability to some nodes, causing them to fall out of predetermined time bounds of the communication scheme, thus forcing them to be almost invisible to the rest of the network. To combat this problem, each node was individually tested.

4.2 Field Tests

TUCCOM's behavior was tested on a number of scenarios, using the setups discussed above as nodes, on the roof garden of the Technical University of Crete's School of Electronic and Computer Engineering. The location is presented in Fig. 4.3. The various test scenarios were divided into 3 categories: a) static topology, b) moving node and c) Interference scenarios. A total of 7 different scenarios were tested using identical test parameters.

In **static topology** tests, each node was randomly placed at a specific location. The network was then let to function until a total number of messages had been successfully sent by each node. In **moving node topology** tests, the network consisted of a statically allocated network and a node, identical to the rest, was moved along the fringes of the network to examine the network behavior under mobility. Every node in the network generated traffic, given by a Poisson distribution with a mean value $\lambda = 1$ message per minute. The goal of each node was to safely forward its data to a sink. The experiments continued until each node in the network successfully transmitted 1000 packets or the moving node transmitted 50 on-demand messages in the moving node tests. The results where gathered by the sink and registered in a connected computer, where they were processed.

(a) Top Down view of a test Configuration, with nodes used, marked.



(b) View of the topology from Fig a, as seen on a lateral level.

Figure 4.3: TUC's roof garden where the experiments where carried out, seen with a sample test scenario.

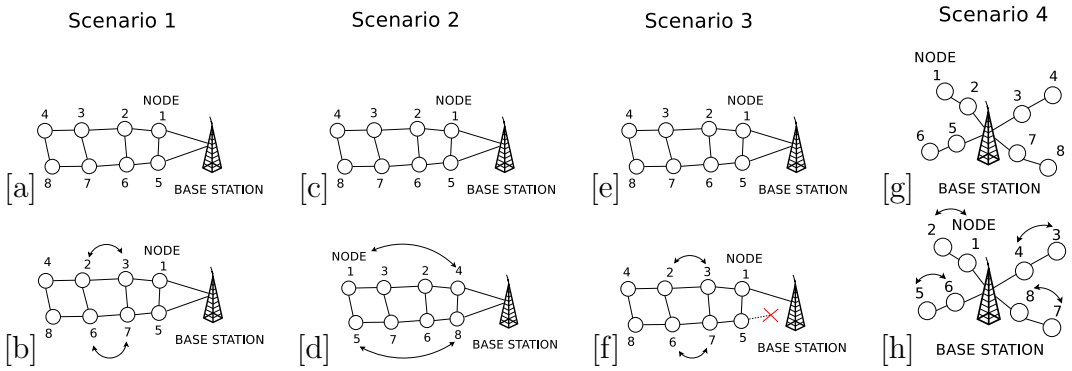
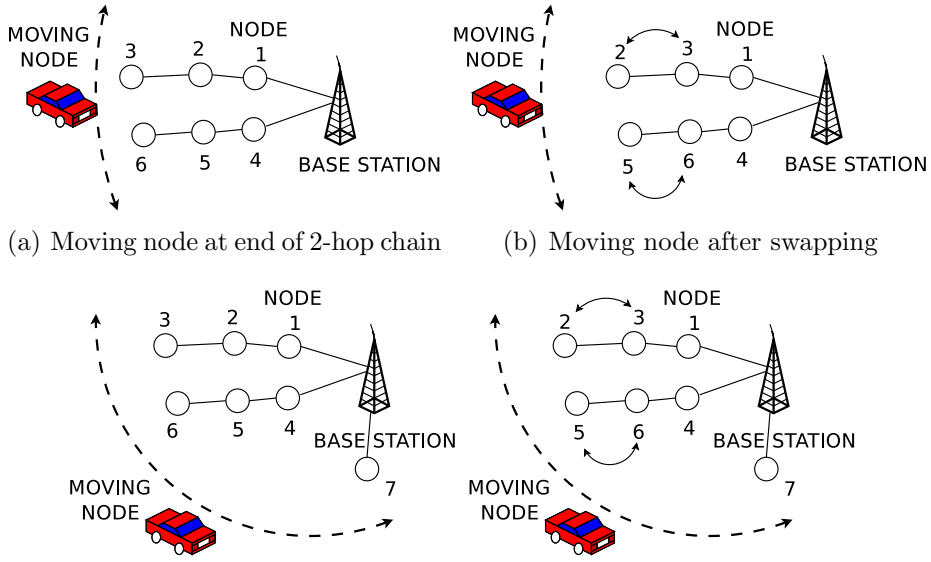


Figure 4.4: Top: Stating 2 4-hop chains, Bot: Network after alterations

4.2.1 Tests under network disruption

To Examine TUCCOM's recovery and robustness features, topology disruption was introduced to the test specifications. In both static and moving node scenarios, the network was initialized and let reach a steady state, then certain nodes had their relative locations swapped. This attempt was done to examine the continuity of communication despite the disruption of initial node-to-sink chains. The specifics of each test category are presented in Tab. 4.1, and the various test topologies themselves in Fig. 4.4-4.5. In moving node scenario's only the results from the moving node are registered, as the rest of the network displays behavior similar to the static topology scenarios.

In these kind of field tests, network operation was as follows. The nodes were randomly booted after being randomly allocated on the field. After they had formed a network with stable routing paths, under standard TUCCOM operation, the node swap disruption was introduced. The point of stable operation was empirically estimated after a series of control tests were carried out; there was no message sniffing infrastructure. After approximately 500 messages from each node were received by the base station, the swapping occurred. The swapping order was random. The network was then let operate until the test was complete and all 1000 messages per node (50 messages from the mobile node, in the appropriate scenario.) were received by the base station. The exact point of equilibrium, in these circumstances, was relatively difficult to derive, thus the before and after swapping results in Tab. 4.4-4.5



(a) Moving node at end of 2-hop chain (b) Moving node after swapping

(c) Moving Node with 3 chains (d) Moving Node with node swap

Figure 4.5: The test setups featuring a moving node.

Table 4.1: Test General Specifications

Scenario	Packets	Power(dBm)	Retransmissions	Generation rate
Static	1000	0 dbm	1	Poisson($\lambda = 1$ min)
Moving	50	0dbm	1	Poisson($\lambda = 1$ min)

are approximate. Each node that successfully transmitted its packets, ended its message generation, but continued foreign traffic propagation.

Data analysis was made possible by the encapsulation of its message statistics in its packets. Each node registered how many messages were lost in link level, how many messages were used for routing setup and maintenance, then encapsulated this information in its message. Thus,, a node level aggregate image exists at the base station were it can be analyzed.

In the interference scenario, a node was programmed to generate random bit streams that served as noise to block a specific channel. The nodes in the interferer's radio range were able to recognize this noise and switched channels of operation to avoid network breakdown. The network, was let form stable links and then the swapping and interference occurred.

Table 4.2: Static Topology results. Before and after disturbance

Scenario	Phase	Avg. Link Accuracy	Overhead (/node)	Avg. Goodput
1	Initialization	95%	40	95%
	After swap	94%	15	87%
2	Initialization	95%	40	95%
	After swap	89%	44	81%
3	Initialization	97%	30	95%
	After swap	90%	21	83%
4	Initialization	98%	30	98%
	After swap	93%	30	95%

4.2.2 Discussion

In Fig. 4.4 the static topology test setups are presented. The top line of each column is the initial layout, while the network after the node interchange is presented at the bottom line. Each column is a separate test scenario, separated into two phases a) initial phase and b) after node interchange (node swapping). In the moving node topology (Fig. 4.5), the top and bottom lines present the layout of scenario 5 and 6, respectively.

The results of the static topology scenarios of Fig. 4.4 are presented in Tab. 4.2. The initial layout of the network for scenarios 1-4 was identical and hence, the results of the initial phase of each scenario are similar (offering a sanity check of the test methodology). In every scenario, a sharp increase in routing messages was observed after node swapping. After detecting unavailability of their former gateways, nodes attempted to route their messages through alternate relays. In all cases after swapping, the RV of neighbors that could serve as these alternate relays became obsolete as they had been also moved. When the nodes decided that their sink access was compromised, they engaged in network rediscovery After new paths were determined, CRIR transmission was stopped and network activity was stabilized. The network in scenario 2 offered the worst performance for the static topology case, while scenario 4 offered the best overall performance. That was due to the importance of the moved node and the length of the rout-

Table 4.3: Moving Node scenario results. Static Nodes Performance

Scen-ario	Phase	Avg. Link Accuracy	Overhead (/node)	Avg. Goodput
5	Init/tion	95%	42	95%
	After swap	94%	17	86%
6	Init/tion	95%	37	95%
	After swap	94%	15	89%

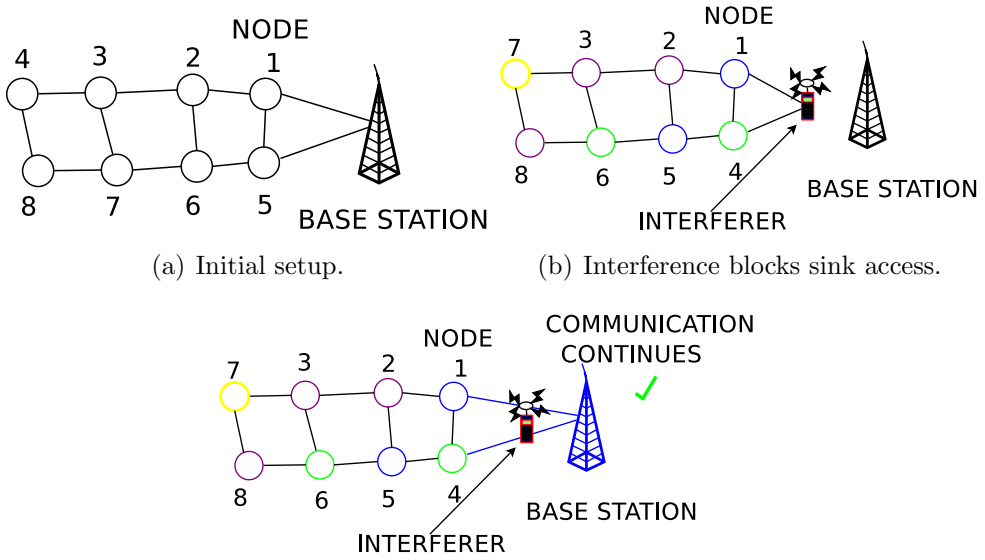
Table 4.4: Moving Node scenario results. Performance of the mobile node

Scen/rio	Phase	Overhead	Goodput
5	Init/tion	36	37%
	After swap	50	31%
6	Init/tion	33	37%
	After swap	42	34%

ing path; in scenario 2, both 1-hop nodes of the network were moved to the end of the routing chain. That caused network disturbance, as nodes had to rediscover their surroundings. In contrast, nodes of scenario 4 offered a more stable reaction due to their star-like topology. The short length of the routing chains allowed them to reestablish reliable paths relatively quickly.

In the moving node topology tests, scenarios 5 and 6 yielded results similar to scenario 1. This can be attributed to the length of the routing chains and the location of the nodes swapped. In scenarios 1, 5, 6 the swapped nodes were located away from the sink, thus of less significance compared to the moved nodes of scenario 2. In scenarios 5 and 6 the static nodes also generated messages, along with propagating the moving node's data. The results presented in Tab. 4.4 describe the mobile node performance. By increasing the impact of the *Badness Commodity*, one timeout was enough to render a specific gateway unreliable. The mobile node would thus attempt to discover new neighbors on each timeout event. That enabled the moving node to retain connectivity.

In the **interference** scenario, a node was employed to jam the main frequency channel. The layout can be seen in Fig. 4.6. Before channel jamming, the network behaved similarly to setup a in Fig. 4.4. After jamming, when 1-



(a) Initial setup.

(b) Interference blocks sink access.

(c) Sink hops to new channel, communication continues

Figure 4.6: Interference scenario setup. Colors denote the different native channels of nodes. In (c), sink hops to another channel (blue color), to avoid interference. The 1-hop nodes, after registering the channel jamming, will now hop to the blue channel to communicate with the sink

Table 4.5: Interference Scenario Results

Scenario	Phase	Avg. Link Accuracy	Overhead (/node)	Avg. Goodput
7	Initialization	95%	40	95%
	After Interference	89%	17	87%

hop nodes 1 and 5 attempted to contact the sink, they encountered elevated interference levels, and successfully migrated to different frequency channels. This introduction of interference induced a loss of 3 messages for nodes 1 and 4 respectively until the channel hopping occurred. Sporadic re-polling of the corrupted channel occurred, due to the tendency of nodes to return to their native channels. These re-polling attempts became increasingly sparse as the channel was found occupied for consecutive times.

4.3 Conclusion

A lightweight, multi-frequency, asynchronous, distributed, interference avoiding MAC-routing protocols was designed, implemented and tested on real nodes. It was able to run on low end nodes, enabling them to form a self organizing, self healing network capable of maintaining functionality despite network alterations and interference. This was done with no central coordination and on unknown topologies with dynamic environments and no hard coded events or behaviors. Nodes displayed a steady behavior and despite limited resources, a simple application was run jointly with the protocol, generating both periodic and event driven traffic. The results implied that a simple, light solution based on a cost function founded on heuristic principles can serve as a basis for ad-hoc network operation, under relatively sparse traffic.

Bibliography

[1] Ian F Akyildiz and Xudong Wang. A survey on wireless mesh networks. *IEEE Commun. Mag.*, 43(9):23–30, September 2005.

[2] Shuguang Cui, Ritesh Madan, Andrea Goldsmith, and Sanjay Lall. Joint routing, mac, and link layer optimization in sensor networks with energy constraints. In *Proc. IEEE Int. Conf. on Commun. (ICC)*, volume 2, pages 725–729, May 2005.

[3] Dario Ferrara, Laura Galluccio, Alessandro Leonardi, Giacomo Morabito, and Sergio Palazzo. Macro: an integrated mac/routing protocol for geographic forwarding in wireless sensor networks. In *Proc. IEEE Int. Conf. on Computer Communications (Infocom)*, volume 3, pages 1770–1781, Miami, USA, March 2005.

[4] Texas Instruments. *CC2500: Low-Cost Low-Power 2.4 GHz RF Transceiver*, 2007.

[5] Sunil Kulkarni, Aravind Iyer, and Catherine Rosenberg. An address-light, integrated MAC and routing protocol for wireless sensor network. *IEEE/ACM Trans. Netw.*, 14(4):793–806, August 2006.

[6] Lucas DP Mendes and Joel JPC Rodrigues. A survey on cross-layer solutions for wireless sensor networks. *Journal of Network and Computer Applications (Elsevier)*, 34(2):523–534, Mar 2011.

[7] Laura Savidge, Huang Lee, Hamid Aghajan, and Andrea Goldsmith. Event-driven geographic routing for wireless image sensor networks. In *Proc. COGIS 06, Paris*, Mar 2006.

[8] Silabs. *C8051F320/1/2/3: USB Microcontrollers Documentation Manual*. Silicon Laboratories, 2003.

[9] Mehmet C Vuran and Ian F Akyildiz. XLP: a cross-layer protocol for efficient communication in wireless sensor networks. *IEEE Trans. Mobile Comput.*, 9(11):1578–1591, November 2010.

[10] Jun Yuan, Zongpeng Li, Wei Yu, and Baochun Li. A cross-layer optimization framework for multicast in multi-hop wireless networks. In *Proc. WICON 05*, pages 47–54, July 2005.

[11] Michele Zorzi. A new contention-based mac protocol for geographic forwarding in ad hoc and sensor networks. In *Proc. IEEE Int. Conf. on Commun. (ICC)*, volume 6, pages 3481–3485, June 2004.