A FRAMEWORK FOR DATA DELIVERY IN INTEGRATED
INTERNET OF THINGS ARCHITECTURES

by

Ashraf E. Al-Fagih

A thesis submitted to the Graduate Program in Computing
in conformity with the requirements for the
Degree of Doctor of Philosophy

Queen’s University
Kingston, Ontario, Canada
April, 2013

Copyright © Ashraf E. Al-Fagih, 2013
Abstract

The Internet of Things (IoT) represents a networking paradigm where entities are viewed as objects that are identifiable, traceable and connected. This view requires the integration and interoperability of numerous wireless standards. Radio Frequency Identification (RFID) systems and Wireless Sensor Networks (WSNs) are two dominant technologies that jointly constitute a class of hybrid/integrated IoT architectures known as RFID-Sensor Networks (RSNs).

Data delivery across such integrated architectures faces challenges in terms of cost-efficiency, scalability and connectivity, among many others. Moreover, IoT-driven solutions are required to address constraints on node mobility, delay-tolerance and resource management, and may have to adhere to an economic model in order to establish incentive-based schemes. Most available RSN solutions are tailored for a single-application and fail to address the aforementioned IoT constraints. To the best of our knowledge, a detailed framework that comprehensively addresses such constraints does not exist. We investigate this promising research direction by proposing a novel framework that incorporates an RSN integrated architecture to improve delivery over heterogeneous topologies. Our framework provides data delivery solutions that adhere to delivery and connectivity considerations of integrated RSN architectures in IoT. Moreover, our data delivery solutions incorporate pricing policies for incentive public sensing applications over the proposed architecture. We show, by theoretical analysis and simulations, that our framework outperforms rival RSN integration approaches, as well as other wireless Ad-hoc data delivery schemes in realizing IoT performance requirements.
Co-Authorship

Chapter 2


Chapter 3


Chapter 4


Chapter 5


To my mother and father.
“The Internet of Things has the potential to change the world, just as the Internet did. Maybe even more so.”

Kevin Ashton, (2009)

“There will always be plenty of things to compute in the detailed affairs of millions of people doing complicated things.”

Vannevar Bush, As We May Think (1945)
Acknowledgements

All praises are to Allah, the most gracious, the most merciful.

This thesis represents the conclusion of a spectacular journey which I was honored to share with a number of amazing individuals.

To my beloved wife Wisam, my life-long partner and soul-mate. This could not have happened if it was not for your limitless love, support and endurance. Your presence has always defined me. I owe every bit of my happiness and success to the beautiful existence of you and our adorable children Faisal and Leena in my life.

To my beloved parents and sisters, your faces were always in my heart. Your prayers were always surrounding me. I know that you are happier with this than I am. I cannot ever repay you.

To my advisor, Dr. Hossam Hassanein, who taught me values beyond mere science and academic practice. Your company over the past years has forever changed me. It definitely made me a better person. Thank you for always being there as a mentor, as a big brother, and as a friend. I’ll always be in your debt, sir.

To my dearest friends, Sharief Oteafy and Fadi Al-Turjman, no words are sufficient to express my graduate to both of you for your pure kindheartedness and genuine devotion.

To my former colleagues at TRL: Abd-Elhamid Taha, Waleed Alsalih, Kashif Ali and Mervat AbuElkheir. Thank you for leaving your permanent marks on my path. It has been an absolute pleasure knowing you.

To all my magnificent friends at TRL and Queen’s, and to the dearest Basia Palmer, thank you for the tremendous love and care you surrounded me with. This story would have had a very different ending if it was not for each and every one of you. I wish you all the best.

To the officials at King Fahd University of Petroleum and Minerals, I express my sincere gratitude for allowing me to go through this endeavor. I look forward towards returning part of the favor to this distinguished establishment.
Statement of Originality

I hereby certify that all of the work described within this thesis is the original work of the author. Any published (or unpublished) ideas and/or techniques from the work of others are fully acknowledged in accordance with the standard referencing practices.

(Ashraf E. Al-Fagih)

(March, 2013)
# Table of Contents

Abstract ............................................................................................................................................ ii
Co-Authorship ................................................................................................................................ iii
Acknowledgements ........................................................................................................................ vii
Statement of Originality ................................................................................................................ viii
List of Acronyms ........................................................................................................................... xv

Chapter 1 Introduction ..................................................................................................................... 1
  1.1 Motivations and Challenges ................................................................................................. 3
  1.2 Thesis Contributions ........................................................................................................... 8
  1.3 Thesis Outline ...................................................................................................................... 10

Chapter 2 Background ................................................................................................................... 11
  2.1 RFID Systems ...................................................................................................................... 11
    2.1.1 Data Delivery in RFID ............................................................................................... 12
    2.1.2 Placement in RFID ................................................................................................... 14
      2.1.2.1 Tag Placement ..................................................................................................... 14
      2.1.2.2 Reader Placement ............................................................................................... 15
  2.2 Wireless Sensor Networks ................................................................................................... 17
    2.2.1 Data Delivery in WSNs ............................................................................................... 19
    2.2.2 Placement in WSNs ................................................................................................... 20
  2.3 RFID-Sensor Networks ....................................................................................................... 22
    2.3.1 Tag-Sensor Integration ............................................................................................... 22
    2.3.2 Reader-Sensor Integration ......................................................................................... 24
    2.3.3 Mixed Integration ...................................................................................................... 25
  2.4 Summary .............................................................................................................................. 27

Chapter 3 Optimized Placement in RSNs ...................................................................................... 28
  3.1 Outlines and Contributions ................................................................................................. 29
  3.2 System Models ..................................................................................................................... 29
    3.2.1 Network Model .......................................................................................................... 30
    3.2.2 Deployment Cost ....................................................................................................... 32
    3.2.3 Communication Model .............................................................................................. 34
  3.3 Placement Strategy ............................................................................................................... 35
  3.4 Experimental Results ......................................................................................................... 39
  3.5 Summary .............................................................................................................................. 43
Chapter 4 Data Delivery ................................................................. 44
  4.1 Outlines and Contributions .................................................. 45
  4.2 Delay-Based Delivery Scheme ............................................. 46
    4.2.1 System Models .............................................................. 48
      4.2.1.1 Network Model ..................................................... 48
      4.2.1.2 Delay Model ....................................................... 51
    4.2.2 Problem Definition and Formulation ............................... 52
    4.2.3 Experimental Results .................................................. 58
      4.2.3.1 Simulation Model ................................................. 58
      4.2.3.2 Simulation Results ............................................... 59
  4.3 Connectivity-Based Delivery Scheme .................................... 63
    4.3.1 System Models .............................................................. 65
    4.3.2 Problem Definition and Formulation ............................... 65
    4.3.3 Experimental Results .................................................. 70
      4.3.3.1 Performance Metrics & Parameters .......................... 70
      4.3.3.2 Baseline Approaches ............................................. 71
      4.3.3.3 Simulation Model ................................................. 71
      4.3.3.4 Simulation Results ............................................... 71
  4.4 Summary .............................................................................. 75

Chapter 5 Priced Data Delivery .................................................... 76
  5.1 Outlines and Contributions .................................................. 78
  5.2 Delivery-Based Approach ................................................... 80
    5.2.1 System Models .............................................................. 80
      5.2.1.1 Network Model ..................................................... 81
      5.2.1.2 Pricing Model ....................................................... 81
      5.2.1.3 Criticalness Model ............................................... 84
      5.2.1.4 Problem Statements and Assumptions ..................... 85
    5.2.2 MCR Heuristics ............................................................. 86
    5.2.3 Experimental Results .................................................. 88
      5.2.3.1 Performance Metrics ............................................. 89
      5.2.3.2 Baseline Approach ............................................... 89
      5.2.3.3 Simulation Results ............................................... 90
  5.3 Sensing-Based Approach .................................................... 94
    5.3.1 Systems models ........................................................... 96
List of Figures

Figure 2.1. RFID tag (left) [40] and reader (right) [41]. .......................................................... 12
Figure 2.2. Reader-tag communication for (a) passive (b) semi-passive and (c) active tags..... 14
Figure 2.3. Multi-hop wireless Sensor Network model. ............................................................. 18
Figure 2.4. Integrated tag-sensor (TS) architecture. ................................................................. 24
Figure 2.5. Integrated RFID reader-sensor (RS) architecture ................................................... 25
Figure 2.6. Mixed integration (MIX) architecture of RFID and WSN. .................................... 26
Figure 3.1. Network model and placement scenarios in SIWR architecture. ......................... 32
Figure 3.2. SIWR cost comparison with 3 SNS/m². ................................................................. 40
Figure 3.3. SIWR cost comparison with 30 SNS/m². ................................................................. 41
Figure 3.4. SIWR cost comparison with 300 SNS/m². ............................................................... 42
Figure 4.1. Network model of DIRSN ...................................................................................... 50
Figure 4.2. Connectivity role of CNs according to DIRSN scheme ......................................... 51
Figure 4.3. Example of DIRSN courier selection. ................................................................. 54
Figure 4.4. Average Delay as CN count increases from 10 to 70 with 10 SNs and 100 LNs/SN. 60
Figure 4.5. Average packet loss against average delay (40 CNs, 10 SNs, 100 LNs/SN) .......... 61
Figure 4.6. Average delay compared average generation rate (40 CNs, 10 SNs, 100 LNs/SN)... 62
Figure 4.7. Partitioning scenario according to second smallest eigenvalue Λ2......................... 64
Figure 4.8. Delay vs. Connectivity for 300 CNs, 20 SNs and 10 LNs/SN. ............................... 72
Figure 4.9. Packet loss vs. Probability of Failure for 300 CNs, 20 SNs and 10 LNs/SN. ......... 73
Figure 4.10. Packet loss vs. Average Generation Rate for 300 CNs, 20 SNs and 10 LNs/SN. .. 74
Figure 5.1. (a) Bottom-up priced delivery and (b) Top-down prised sensing ......................... 78
Figure 5.2. Network model for MCR ....................................................................................... 81
Figure 5.3. Energy consumed vs. pause time for 100 SNs and 900 CNs................................. 91
Figure 5.4. Energy consumed vs. pause time for 500 SNs and 500 CNs ................................. 92
Figure 5.5. Delivery fraction vs. pause time for 100 SNs and 900 CNs ..................................... 93
Figure 5.6. Delivery fraction vs. pause time for 500 SNs and 500 CNs ..................................... 94
Figure 5.7. Architectural model of priced public sensing (PPS) in IoT. ............................... 96
Figure 5.8. Components of PPS Network model. .................................................................. 97
Figure 5.9. Illustration of two data collecting scenarios (a) SDC (b) MDC. .......................... 99
Figure 5.10. Delay function plot. ......................................................................................... 109
Figure 5.11. Quality function plot. ....................................................................................... 110
Figure 5.12. Trust function plot. ......................................................................................... 111
Figure 5.13. Delay vs. Quality 3-D rendering ................................................................. 112
Figure 5.14. Trust vs. Delay 3-D rendering ................................................................. 113
Figure 5.15. Trust vs. Quality 3-D rendering ($\sigma = 2$) ........................................ 114
Figure 5.16. Utility Vs. Delay for different data types .................................................. 115
Figure 5.17. Utility Vs. Quality for different data types .............................................. 116
Figure 5.18. Average packet delivery vs. LN count (100 GWs, 5-30 DCs) ............... 125
Figure 5.19. Average packet delivery vs. DC count (100 GWs, 4500 LNs) ............... 126
Figure 5.20. Average packet delivery vs. pause (100 GWs, 5-30 DCs, 4500 LNs) .... 127
Figure 5.21. Average delay vs. Sensing field radius ($R$) (100 GWs, 5-30 DCs, 4500 LNs) ...... 128
Figure 5.22. Average network delay vs. DC counts (100 GWs, 4500 LNs) .......... 129
Figure 5.23. Average network delay vs. pause time (100 GWs, 5-30 DCs, 4500 LNs) ...... 130
Figure 5.24. Average network lifetime vs. $D$ (pkt/round) (100 GWs, 5-30 DCs, 4500 LNs) .... 131
Figure 5.25. Average network lifetime vs. DC count (100 GWs, 4500 LNs) .............. 132
Figure 5.26. Average network lifetime vs. pause time (100 GWs, 5-30 DCs, 4500 LNs) ...... 133
Figure 5.27. Average delay vs. Price increment percentage (100 GWs, 5-30 DCs, 4500 LNs) . 134
Figure 5.28. Average network lifetime vs. Price increment (100 GWs, 5-30 DCs, 4500 LNs) . 135
## List of Tables

Table 1.1. Comparative summary of RFID, WSN and RSN systems............................................. 4  
Table 2.1. Comparative summary of planned RFID reader placement schemes ......................... 17  
Table 2.2. Summary of RSN integration architectures. ................................................................. 27  
Table 3.1. ILP formulation for SIWR........................................................................................... 38  
Table 4.1. ILP formulation for DIRSN......................................................................................... 56  
Table 4.2. SDP formulation for URIA.......................................................................................... 68
**List of Acronyms**

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGR</td>
<td>Average Generation Rate</td>
</tr>
<tr>
<td>AND</td>
<td>Average Network Delay</td>
</tr>
<tr>
<td>ANL</td>
<td>Average Network Lifetime</td>
</tr>
<tr>
<td>AODV</td>
<td>Ad-hoc On-demand Distance Vector</td>
</tr>
<tr>
<td>AP</td>
<td>Access Point</td>
</tr>
<tr>
<td>APD</td>
<td>Average Packet Delivery</td>
</tr>
<tr>
<td>APL</td>
<td>Average Packet Loss</td>
</tr>
<tr>
<td>BAN</td>
<td>Body Area Network</td>
</tr>
<tr>
<td>BS</td>
<td>Base Station</td>
</tr>
<tr>
<td>CN</td>
<td>Courier Node</td>
</tr>
<tr>
<td>CSMA/CD</td>
<td>Carrier Sense Multiple Access with Collision Detection</td>
</tr>
<tr>
<td>DC</td>
<td>Data Collector</td>
</tr>
<tr>
<td>DIRSN</td>
<td>Delay-tolerant approach for Integrated RFID-Sensor Networks</td>
</tr>
<tr>
<td>DSD</td>
<td>Delay-Sensitive Data</td>
</tr>
<tr>
<td>DSR</td>
<td>Dynamic Source Routing</td>
</tr>
<tr>
<td>DTD</td>
<td>Delay-Tolerant Data</td>
</tr>
<tr>
<td>DTN</td>
<td>Delay-Tolerant Network</td>
</tr>
<tr>
<td>EPC</td>
<td>Electronic Product Code</td>
</tr>
<tr>
<td>FTRN</td>
<td>Fast-Track Relay Node</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GV</td>
<td>Geographic Vicinity</td>
</tr>
<tr>
<td>GW</td>
<td>Gateway</td>
</tr>
<tr>
<td>HRM</td>
<td>Hard Real-time bounded Multimedia</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>ILP</td>
<td>Integer Linear Programming</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>LN</td>
<td>Light Node</td>
</tr>
<tr>
<td>MANET</td>
<td>Mobile Ad-hoc Network</td>
</tr>
<tr>
<td>MCR</td>
<td>Monetary-based Courier Relaying</td>
</tr>
<tr>
<td>MDC</td>
<td>Mobile Data Collector</td>
</tr>
<tr>
<td>MIX</td>
<td>Mixed Integration</td>
</tr>
<tr>
<td>NRRA</td>
<td>New Reliable Routing Algorithm</td>
</tr>
<tr>
<td>OLSR</td>
<td>Optimized Link State Routing</td>
</tr>
<tr>
<td>ORP</td>
<td>Optimized Relay Placement</td>
</tr>
<tr>
<td>OV</td>
<td>Optimized Value</td>
</tr>
<tr>
<td>P2P</td>
<td>Peer-to-peer</td>
</tr>
<tr>
<td>PODV</td>
<td>Peripheral On-demand Distance Vector</td>
</tr>
<tr>
<td>PoF</td>
<td>Probability of Failure</td>
</tr>
<tr>
<td>PPS</td>
<td>Priced Public Sensing</td>
</tr>
<tr>
<td>PRRA</td>
<td>Peripheral Reliable Routing Algorithm</td>
</tr>
<tr>
<td>PS</td>
<td>Public Sensing</td>
</tr>
<tr>
<td>PSN</td>
<td>Pocket Switched Networking</td>
</tr>
<tr>
<td>QoE</td>
<td>Quality of Experience</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
<tr>
<td>RFID</td>
<td>Radio Frequency IDentification</td>
</tr>
<tr>
<td>RN</td>
<td>Relay Node</td>
</tr>
<tr>
<td>RS</td>
<td>Reader-Sensor (Integration)</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>RSN</td>
<td>RFID-Sensor Network</td>
</tr>
<tr>
<td>SCF</td>
<td>Store-Carry-Forward</td>
</tr>
<tr>
<td>SDC</td>
<td>Static Data Collector</td>
</tr>
<tr>
<td>SDP</td>
<td>Semi-Definite Programming</td>
</tr>
<tr>
<td>SIWR</td>
<td>Smart Integrated WSN-RFID</td>
</tr>
<tr>
<td>SN</td>
<td>Super Node</td>
</tr>
<tr>
<td>SRM</td>
<td>Soft Real-time bounded Multimedia</td>
</tr>
<tr>
<td>TS</td>
<td>Tag-Sensor (Integration)</td>
</tr>
<tr>
<td>URIA</td>
<td>Ubiquitous Robust Integrated Approach</td>
</tr>
<tr>
<td>WiFi</td>
<td>Wireless Fidelity</td>
</tr>
<tr>
<td>WSN</td>
<td>Wireless Sensor Network</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

The Internet of Things (IoT) has gained rapid attention as a comprehensive paradigm that is driven by an expansion of the Internet. The phrase “Internet of Things” is believed to have originated at the beginnings of the century in correspondence to the work done at the MIT Auto-ID Center [1] to develop industry-oriented identification technologies to automate, reduce errors and increase efficiency [2]. However, the IoT model has grown since to accommodate any object capable of interacting directly with its local neighbours. In this context, the Internet can be viewed as a backbone network that interconnects a huge number of smaller (peripheral) networks, each of which would regroup objects according to its neighborhood relationships and physical properties. Examples of such smaller networks include sensor networks, vehicular networks and Mobile Ad-hoc Networks (MANETs) in general.

The IoT will have a tremendous effect on all aspects of everyday life, promising to eventually provide identification, tracking and communication abilities to virtually every object on the planet [3]. The IoT will revolutionize networking over a myriad of applications, including participatory sensing, enhanced learning, e-health and automotive applications. Similarly, IoT’s influence will reform numerous business disciplines such as intelligent manufacturing, retail, supply chains and product lifecycle management, in addition to reliable and safe transportation of people and goods [2], [4], [5], [6], [7], [8].

The Interconnection of the abundance of communication standards residing within the IoT via hybrid/integrated architectures, to form a large-scale, heterogeneous and distributed network of objects, is the core of the IoT vision and its most pressing difficulty. In this regard, Radio...
Frequency Identification (RFID) systems and Wireless Sensor Networks (WSNs) represent the most dominant and promising technologies [1], [9], [10], [11].

RFID systems are predominantly comprised of tags (with unique IDs) that communicate with a central reader. This master-slave relationship dominates the RFID systems paradigm [12]. By attaching these tags to objects, the RFID system enables rapid and cost-effective methods of object identification and tracking. In fact, earlier visions of IoT considered things as simple as RFID tags [1]. The first efforts to realize the IoT were driven by attempts to support industry standards that utilize RFID tags in world-wide trading networks. Such standards were designed to improve object visibility (i.e. the traceability of an object and the awareness of its status, current location, etc.) [5]. Eventually, the IoT grew to exceed the simple identification capacities of RFID. It is widely accepted that Wireless Sensor Networks (WSNs) represent the most prominent technology to produce the backbone of IoT along with RFID. The two technologies jointly form what is known as RFID-Sensor Networks (RSNs) [5], [13], [14], [15], [16].

Traditionally, WSNs are viewed as dense deployments of small low-powered sensor nodes with restricted computational, communication and battery resources. Sensor nodes monitor physical phenomena such as temperature, motion and levels of contamination and report to specific base stations or access-points that are connected to the Internet. WSNs are employed in environment monitoring, biomedical observation, surveillance and security, among a multitude of domains [17], [18], [19]. However, the dominantly narrow view of WSNs fails to encompass a wider set of topologies, including mobile and vehicular networks where sensors onboard vehicles and pervasive devices form sensory networks that are more advanced in terms of mobility, buffering, computational and communicational capacities. We note that such pervasive devices are considered as essential components of the envisioned IoT paradigm. Hence, we adopt through
the remainder of this thesis an expanded view of a WSN to incorporate systems of heterogeneous cost-effective devices including handheld devices, medical sensors, smartphones, tablet PCs and relays that vary in their power, computational capacity and communication ranges. This large body of pervasive mobile “sensory” devices, in addition to RFID and WSN components represents the building blocks of our RSN-based architecture.

1.1 Motivations and Challenges

The integration of RFIDs and WSNs in RSNs will supersede their independent capabilities and enable a plethora of IoT applications. An RSN node may draw its energy from the electromagnetic field of the RFID reader, which reduces the power consumption to a minimum and extends its lifetime significantly. In addition, RFID tagging provides a wider range of addresses than the traditional IP addressing, with current standards allowing up to $2^{128}$ addresses [11]. Moreover, RFID tagging enables WSNs to track objects that otherwise are difficult to sense and locate. WSNs, on the other hand, add intelligence to RFIDs by providing more advanced processing and buffering resources, in addition to detailed information on external parameters such as temperature, humidity, etc. Furthermore, while RFID is limited to single-hop reader-tag communication, WSN nodes can inter-communicate over multi-hops in contrast to the reader’s range of interrogation. The main characteristics and differences of RFIDs, WSNs and RSNs are summarized in Table 1.1.

Integrating heterogeneous components and designing efficient data delivery schemes for their unified network model, presents various challenges particularly in terms of interoperability, placement and cost-efficiency. Having both RFID and WSN technologies to operate concurrently implies additional costs related to designing and deploying integrated hardware components.
<table>
<thead>
<tr>
<th><strong>Use</strong></th>
<th><strong>RFIDs</strong></th>
<th><strong>WSNs</strong></th>
<th><strong>RSNs</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Identify and locate tagged objects</strong></td>
<td>Possibility of supporting sensing, computing and communication capabilities in a passive system</td>
<td>Conduct both RFID and WSN purposes</td>
<td></td>
</tr>
<tr>
<td><strong>Components</strong></td>
<td>Tags, readers</td>
<td>Sensor node, relay node, base station</td>
<td>Sensor/tags, readers, base station and mobile transceivers</td>
</tr>
<tr>
<td><strong>Communication</strong></td>
<td>Single-hop</td>
<td>Multiple-hop</td>
<td>Require both relays and readers to support multi-sink nature</td>
</tr>
<tr>
<td><strong>Common Standard</strong></td>
<td>EPC standard</td>
<td>IEEE 802 family (ZigBee, WiFi, Bluetooth, etc.)</td>
<td>In progress</td>
</tr>
<tr>
<td><strong>Range</strong></td>
<td>Few centimeters to few meters</td>
<td>Dependent on transceiver properties, typically 10’s to 100’s of meters</td>
<td>Capped by that of RFID readers and/or WSN relays</td>
</tr>
<tr>
<td><strong>Power</strong></td>
<td>Active, passive or semi-passive</td>
<td>Active with on-chip battery</td>
<td>Active, using WSN battery and harvested power</td>
</tr>
<tr>
<td><strong>Addressing</strong></td>
<td>Tag ID</td>
<td>MAC dependent, most recently IPv6 enabled</td>
<td>Support a dual tag ID/MAC addressing system</td>
</tr>
<tr>
<td><strong>Size</strong></td>
<td>Very small tags</td>
<td>Medium to small sensors</td>
<td>Vary according to integration purpose</td>
</tr>
<tr>
<td><strong>Cost</strong></td>
<td>Tag: very inexpensive Reader: expensive</td>
<td>Sensor node: inexpensive Relay node: expensive</td>
<td>Sensors/tags: inexpensive Relay/Reader: expensive couriers are ubiquitously available user nodes that are paid per use</td>
</tr>
<tr>
<td><strong>Mobility</strong></td>
<td>Move with attached object</td>
<td>Usually static</td>
<td>Varied mobile</td>
</tr>
<tr>
<td><strong>Placement</strong></td>
<td>Mostly fixed</td>
<td>Random or deterministic</td>
<td>Adapt to couriers’ trajectory</td>
</tr>
</tbody>
</table>
These components are naturally more complicated than simple sensors or tags. In case of RFID deployment, particularly, redundant readers’ deployment is a common practice to guarantee area coverage. Redundancy, however, is not a cost-effective approach and poses the side effect of creating significant interference among readers and consequently degrading the performance of the whole system. Nonetheless, overlaps among readers’ zones represent an inevitable consequence of achieving full coverage.

IoT assumes that things, being physical or virtual, have digital functionalities and can be identified and tracked automatically by their corresponding neighbors in order to implement a multitude of data delivery schemes. This assumption has critical implications in terms of the connectivity, deliverability and deployment aspects of IoT architectures, which by themselves represent a concept yet to be fully defined and realized [20].

Unfortunately, most of the RSN integrated architectures proposed in the literature are application-specific and do not cater for ultra-large-scale deployments in their integration approaches, which reflects on the poor level of cost efficiency they achieve when considered for IoT-driven deployment. Such architectures as well fail in utilizing ubiquitously available components in IoT environments.

An inherent characteristic in IoT is delay-tolerance [21]. In reality, an IoT node has only partial knowledge regarding the full path to the destinations assigned to the packets it carries. Due to partitioning, which is mainly caused by nodal mobility, connectivity may occur on an intermittent basis. In such situations, nodes are required to store and carry the packets until a suitable forwarding opportunity arises in a Store-Carry-Forward (SCF) delivery fashion [22]. Traditional WSN protocols are inherently mobility-intolerant since most of the WSN network architectures assume stationary sensor nodes [23]-[30]. As stated earlier, we adopt an expanded notion of
sensor networks that incorporates MANET nodes. An abundance of routing-layer protocols have been proposed to accommodate the dynamic topology in MANETs [31]-[34]. Yet, for all these protocols, it is implicitly assumed that the network is connected and there is a contemporaneous end-to-end path between any source/destination pair. In other words, the topology in the standard dynamic routing problem is assumed to be always connected and the objective of the routing algorithm, hence, is confined to finding the best currently available full path to move traffic from one end to the other. Unfortunately, none of these assumptions stand in a delay-tolerant setup. An IoT data delivery scheme must be delay-tolerant to cope with intermittent connectivity, in addition to providing faster delivery alternatives for other delay-sensitive types of data that demand minimal delays.

Vehicles and handheld gadgets equipped with advanced buffering, processing and transceiving capabilities will play a major role as communication facilitators in the IoT paradigm. Such pervasive mobile nodes topologies (labeled onwards as couriers) close the gap between the real-world and its virtual representation via seamless identification and integration with other wirelessly-embedded devices and their surroundings.

Courier nodes may be utilized as delay-tolerant linkers. But their services cannot be taken for granted. The notion of creating incentive for wireless nodes to take part in a group task has recently gained prominence in the wireless networks literature [35], [36]. This fits the IoT’s propensity to provide large-scale information access. One particularly promising model in this regard is public (or participatory-centric) sensing (PS) which develops large-scale sensor networks at low cost by utilizing everyday sensory and mobile devices in applications where data is shared among users for the greater public good [37], [38]. Such a comprehensive public sensing vision is more diverse and complicated than RSNs are, and dictates broadening the set of
services provided to potential clients, the data delivery schemes, and the economic growth in the system as a whole.

Pricing schemes in heterogeneous networks cover a broad spectrum of factors to determine the value of a resource. In general, the most efficient schemes capitalize on the differential values of each of the constituents. Following the basic laws of supply and demand, the abundance of resources and their homogeneity decrease their value. However, higher prices are usually assigned to nodes with scarce services, or those with elevated Quality of Service (QoS) measures[36]. Generally, factors such as bandwidth, buffering capacity, residual energy and tendency to assume selfish behavior in the network, all contribute to pricing.

To this end, we define an Internet of Things Setting by the following four main characteristics: 1) Ability to identify 2) Seamless integration 3) Ubiquitous & robust connectivity and 4) Delay-tolerance. Hence, we provide a framework, encompassing a cost-efficient RSN architecture, to address data delivery and placement objectives according to the aforementioned characteristics of the IoT setting. The design objectives are to be met with respect to metrics such as delay-tolerance, connectivity and lifetime. Our proposed framework will provide identification and sensing services provided by RFID and WSN technologies as part of an integrated architecture. In addition, this architecture will make use of ubiquitous couriers available in today’s topologies to enhance connectivity and delivery rates between the components of the integrated topology. Our framework will as well provide delivery guarantees with respect to delay and connectivity over end-to-end links. Such guarantees will be carried out by dedicated components of our integrated architecture, in addition to other components incorporated within the wider IoT vision.
The objective of this thesis is to provide a data delivery framework for RSN architectures that address cost efficiency, delay, connectivity and pricing requirements. The characteristics of our data delivery framework are summarized as follows:

1) Cost-effective. Minimally deploying expensive components to control cost factor.
2) Delay-tolerant. In case of a partitioning, nodes are capable of buffering messages until a suitable forwarding opportunity arises.
3) The framework must exploit ubiquitously available couriers that are neither sensor nodes nor RFID tags, such as vehicles and handheld devices equipped with buffering and transmission capabilities.
4) It satisfies connectivity constraints by maintaining a minimum number of end-to-end connections for a given layout.
5) It adheres to a pricing scheme for courier utilization, following the basic laws of supply and demand, in addition to catering for user QoS requirements within a PS paradigm.

1.2 Thesis Contributions

Our research aims primarily at providing optimized placement and data delivery solutions for integrated RSN architectures in IoT by exploiting the mobility of ubiquitous courier nodes. Our work is of a progressive nature in the sense that each presented part builds on its predecessor. The main contributions of this thesis are the following:

1) We introduce a novel architectural approach for RSNs. The functional and cost efficiencies of this architecture, however, are dependent on the accurate placement of its most complex components. Hence, we provide the Integer Linear Programming (ILP) formulation for optimal placement of the minimal number of such components while guaranteeing connectivity and coverage constraints.
2) We incorporate ubiquitous or pervasive mobile nodes common to IoT environments into our RSN architecture as couriers with linking tasks between partitioned parts of the network.

3) We introduce a delay-based delivery scheme for our RSN architecture. The scheme employs a novel ILP-based formulation that guarantees minimal delay across the integrated topology while obeying link-capacity and load balancing constraints.

4) We introduce an alternative connectivity-based delivery scheme which utilizes the mobility of couriers to guarantee a specific level of connectivity across the RSN topology. This scheme incorporates a Semi-Definite Program (SDP) formulation to achieve a guaranteed connectivity level, in addition to adhering to delay constraints for applications that are not delay-tolerant.

5) We introduce online heuristics detailing the mechanism of dynamic assignment of relaying pricing for courier nodes. Our heuristics allow, on the one hand, data sources to decide on the price they could afford to transmit the data based on a data criticalness function. On the other hand, the heuristics allow each courier to arbitrarily decide its current “charge” for forwarding a data packet to a specific destination. Our pricing model caters for heterogeneity stemming from transmission limitations of couriers.

6) We propose a comprehensive network model and pricing scheme for public sensing applications based on IoT-driven topologies that integrate heterogeneous data sources in an infrastructure-less environments. We provide a dynamic multi-tier pricing scheme that, from the suppliers’ end, adheres to the social welfare of the public sensing system by incorporating lifetime and energy constraints while considering, from the consumers’ end, delay and quality attributes of the received data to insure maximum utility gain. Our
public sensing scheme includes heuristics specifying a distributed data delivery approach that exploits the components of the aforementioned architectural model, including stationary and mobile data sources, in order to satisfy different delay requirements according to the pricing constraints.

1.3 Thesis Outline
The remainder of this thesis is organized as follows. Chapter 2 highlights relevant background and related work in the literature in terms of RFID systems, WSNs and RSN integration approaches. In Chapter 3, we present our novel integration approach and an ILP formulation for optimal placement of integrated nodes to achieve efficiency in terms of deployment cost. In Chapter 4, we investigate two approaches of data delivery in RSN architectures. First, we propose a delay-based approach that guarantees minimal delay connections across the integrated topology by executing an ILP-based solution for best courier selection. Second, we propose a connectivity-based approach that guarantees a specific level of connectivity by executing an SDP-based algorithm. In Chapter 5, we improve our data delivery schemes by adding economic considerations. We propose two approaches for priced data delivery in IoT-driven applications. The first approach follows a bottom-up pricing model where lower-tier components of our architecture pay to get their data delivered to destination at the top of the hierarchy. The delivery price is determined here according to a data criticalness function and resource abundance such as buffer space and energy. The second priced scheme is top-down and is based on public sensing scenarios, where upper-tier entities initiate data purchasing requests. Prices are paid to lower-tier entities according to quality constraints such as delay and trust. Lastly, Chapter 6 concludes this thesis and outlines some future research directions.
Chapter 2

Background

This Chapter presents the background material and surveys previous research related to the main components and concepts of the framework presented in this thesis. Section 2.1 provides an introduction to RFID systems and their role in the IoT paradigm, with two subsections surveying placement approaches for RFID tags and readers, respectively. Section 2.2 provides a background on WSNs in terms of components, delivery schemes and placement approaches. Section 2.3 overviews the main RSN integration architectures. A brief summary is provided in Section 2.4.

2.1 RFID Systems

A Radio Frequency IDentification (RFID) system consists of three main components: 1) a reader (transceiver) which may be a read or write/read device, 2) a tag (transponder) and 3) the communication between them. Additional components may also include antennas that emit radio signals to activate the tag and read/write data to it, and a local control chamber sending reading/writing commands to all the readers as well as reading back tag information [12]. The most representative RFID management system is the EPC global network developed by EPCglobal [39].

The RFID reader is the more complex unit. Readers’ cost and level of deployment mostly reflect that of the system as a whole. Tags, on the other hand, are usually mass produced which contributes to their lower cost compared to that of a reader. Each tag has an identification (ID) number and a memory that stores additional data such as manufacturer, product type, etc.
RFID has several advantages over traditional identification technologies (e.g. barcodes) and influences many application domains including inventory control and supply chain management [42]. RFID tags may serve as personal data recording devices in addition to providing positional accuracy that may surpass that of GPS especially in indoor settings [43]. Most importantly, RFID systems stand at the forefront of the technologies driving the IoT vision [12]. They are favored because of their non-disruptive small size, low cost and extended lifetime.

2.1.1 Data Delivery in RFID

RFIDs apply radio frequency (RF) electromagnetic fields to transfer data from tags to readers for the purposes of identification and tracking tagged objects. Tags are classified into: 1) passive 2) semi-passive or 3) active according to their power supply and transmission approaches. Passive and semi-passive RFID tags transmit their data by reflection or modulation of the electromagnetic field emitted by the reader. Passive tags, however, do not have a power supply of their own and gain their energy from the reader’s field within an interrogation zone ranging
between 10 cm and 3 m. Common applications of passive RFID tags include credit cards and electronic door keys. Semi-passive tags operate similarly to passive tags. However, their circuitry includes a battery enabling them to transmit their data once interrogated by a reader for distances longer than passive tags. Data loggers which can take sensor readings automatically while running real-time clocks are common utilizers of semi-passive tags.

Conversely, an active RFID tag self-broadcasts its data for extended communication ranges of up to 100 m without the need for an interrogating reader. Since active communication requires larger batteries and more advanced circuitry, the typical price of active tags is five to ten times the price of semi-passive RFID tags [44]. Each of the tag types has a separate range of transmission frequency band. Passive tags, for instance, usually transmit at a low frequency (120-150 kHz), while semi-passive and active tags use high (13.56 MHz) and ultra-high frequency (433 MHz) bands [12]. A depiction of RFID reader-tag communication according to tag type is shown in Figure 2.2.

The number of tags within each reader’s interrogation zone determines the interrogation delay of that reader. The overall interrogation delay of the system is determined by the longest delay associated with a single reader. Therefore, reducing overlaps and balancing the load of readers, in addition to reducing the delay and monetary cost of the system, are all considerable constraints for any efficient RFID system.
2.1.2 Placement in RFID

Placement (i.e. choosing the optimum locations for readers, tags or both) is of utmost importance for a variety of considerations. Some RFID systems achieve a maximum coverage of 90% of the intended area without overlaps due to the nature of electromagnetic readers’ fields of communication [45]. In the following subsections, we provide an overview of placement approaches for RFID tags and readers, respectively.

2.1.2.1 Tag Placement

Tag placement may be broadly classified into either dynamic or static. Examples of dynamic placement include tags assigned to individuals moving across a given workplace or a city-section, or transported items to be tracked by real-time inventory and supply chain management solutions [42]. As for static tag placement, its applications include defining boundaries within
buildings for security measures or to assist the visually-impaired. In [46], for instance, an intelligent system is proposed to automatically suggest the tag placement locations and calculate the number of tags required for implementing an indoor navigation system for the visually-impaired. The reader, in this scenario, is carried by the visually-impaired person. Authors in [47] introduce a tagging system for 3-D location sensing based on radio signal strength. Such systems serve for both positioning and tracking purposes, and are more reliable than Global Positioning Systems (GPS) since they operate indoors and do not require an expensive network of satellites.

2.1.2.2 Reader Placement

RFID reader placement is more complex in terms of cost and coverage constraints. Hence, more extensive studies have been conducted in this regard as detailed in the following.

The two broader reader placement approaches are controlled (planned) and random (Ad-hoc). Controlled reader placement is usually pursued for indoor [48] and city-scape applications [49] where tagged items are expected to follow a high level of predictability in their mobility, as opposed to tags in environmental applications, for example. The aforementioned approaches apply algorithms to find an optimal placement of RFID readers in a 2-D grid [48], [50], [51], [52], or in 3-D set-cover space [53]. These placements are mostly coverage-oriented [45], [51], [53], [54]. Other controlled reader placement objectives include tracking [48], [49], and positioning [52], [55]. In the survey we conducted [56], we pointed to these criteria in addition to proposing a classification of coverage schemes depending on the extent of the readers’ coverage zone as 3-D (space), 2-D (plane), 1-D (linear) or point coverage. Examples of linear coverage include defining street routes or room boundaries [45], [49]. A common point coverage example is electronic door key-reader for access control applications.
Some deterministic approaches utilize algorithms to find an optimal placement of RFID readers in a grid [51], or at a predetermined set of locations [57]. While these schemes are able to achieve maximal coverage, they suffer from two major weaknesses when applied to integrated RSN layouts. First, the coverage comes at the heavy price of an abundance of expensive integrated units. The grid approach is particularly artificial for real-life IoT scenarios where node locations are not proprietary but rather dependent on active/human mobility. If the grid approach is to be applied with the RS integrated architecture, for instance, we will end up with a situation where tags are distributed in a grid fashion.

The random reader placement approach, on the other hand, is generally applied for item-level tagging applications [58]. Yet, Ad-hoc reader placement results in the intersection of their electromagnetic interrogation zones, which is known as reader-to-reader collision. This phenomenon typically invokes running redundancy elimination algorithms to turn off redundant readers that are not covering any tag in the vicinity in order to improve the system’s performance and lifetime. Alas, the redundant reader elimination problem is known to be NP-hard. Numerous schemes have been proposed to solve this problem [59], [60], [61], [62]. The aforementioned reader-placement proposals are summarized in Table 2.1.
Table 2.1. Comparative summary of planned RFID reader placement schemes.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Type</th>
<th>Topology</th>
<th>Cvg. Space</th>
<th>Coverage Method</th>
<th>Tag Type</th>
<th>Objective</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>[45]</td>
<td>Mobile</td>
<td>Zigzag</td>
<td>1-D</td>
<td>Tag-based</td>
<td>Passive</td>
<td>Full coverage</td>
<td>Retail inventory tracking</td>
</tr>
<tr>
<td>[48]</td>
<td>Static</td>
<td>Square</td>
<td>2-D</td>
<td>Area-based</td>
<td>Passive</td>
<td>Tracking and positioning</td>
<td>Indoor tracking</td>
</tr>
<tr>
<td>[49]</td>
<td>Static</td>
<td>City-scape</td>
<td>1-D</td>
<td>Tag-based</td>
<td>Passive</td>
<td>Tracking</td>
<td>Vehicular networks</td>
</tr>
<tr>
<td>[50]</td>
<td>Static</td>
<td>Hexagonal grid</td>
<td>2-D</td>
<td>Area-based</td>
<td>--</td>
<td>Minimize collisions</td>
<td>--</td>
</tr>
<tr>
<td>[51]</td>
<td>Static</td>
<td>Square</td>
<td>2-D</td>
<td>Area-based</td>
<td>Active</td>
<td>Full coverage</td>
<td>Asset tracking in hospitals</td>
</tr>
<tr>
<td>[52]</td>
<td>Static</td>
<td>Octagonal</td>
<td>2-D</td>
<td>Area-based</td>
<td>Active</td>
<td>Accurate positioning</td>
<td>Indoor location sensing</td>
</tr>
<tr>
<td>[53]</td>
<td>Static</td>
<td>--</td>
<td>3-D</td>
<td>Area-based</td>
<td>Passive</td>
<td>Full coverage</td>
<td>Generic</td>
</tr>
<tr>
<td>[54]</td>
<td>Static</td>
<td>--</td>
<td>2-D</td>
<td>Area-based</td>
<td>Passive</td>
<td>Full coverage</td>
<td>--</td>
</tr>
<tr>
<td>[55]</td>
<td>Static</td>
<td>Grid</td>
<td>2-D</td>
<td>Tag-based</td>
<td>Passive/Active</td>
<td>Accurate positioning</td>
<td>Pinpointing misplaced tags</td>
</tr>
<tr>
<td>[57]</td>
<td>Static</td>
<td>--</td>
<td>3-D</td>
<td>Tag-based</td>
<td>Passive</td>
<td>Maximize powering region</td>
<td>Supply chain management</td>
</tr>
</tbody>
</table>

2.2 Wireless Sensor Networks

Traditionally, Wireless Sensor Networks (WSNs) are viewed as dense deployments of tiny low-powered sensor nodes with restricted computational, communicational and power capabilities [14]. Sensor nodes read physical phenomena such as temperature, motion and levels of contamination and use multi-hop communication to report their readings, usually via relay nodes, to specific base stations or access points that are connected to the Internet; Figure 2.3 depicts this WSN model.
WSN deployments may also include a wider set of contemporary topologies, viz. Mobile Ad-hoc Networks (MANETs), where sensors onboard vehicles and ambient personal devices (e.g. smartphones) form sensory networks that are more advanced in terms of mobility, buffering, processing and transmission capacities. We note that such ambient devices are considered to be core components of our envisioned IoT paradigm. Hence, we adopt such expanded view of a WSN that includes systems of heterogeneous cost-effective devices such as sensors, cellular phones, tablet-PCs and relay nodes that vary in their functional capabilities. These devices are expected as well to apply a variety of wireless standards including WiFi, Bluetooth, Zigbee and Body Area Networks (BANs).

In a typical WSN topology, if every node communicates directly with the access point, regardless of its position in the network, then the communication load would exhaust the system’s power resources. Thus, sensors operate in a decentralized multi-hop fashion in order to maintain...
connectivity while extending the system’s lifetime such that a subset of sensors is responsible for relaying the aggregate data to access points. This communication model can be generalized under the umbrella of IoT to include nodes resourceful of energy supplies for considerations related to pricing and cost-efficiency.

Since our IoT framework focuses on data delivery and placement, we survey each of these WSN approaches that cover these two concepts in the following subsections.

2.2.1 Data Delivery in WSNs

Data delivery in WSNs is application dependent. The data delivery model from the sensor to the access point can be classified as either: continuous, event-driven, query-driven or hybrid [17], [22], [24]. In the continuous model, each sensor sends data periodically. In event-driven and query-driven models, the transmission of data is triggered respectively when an event occurs or a query is generated by the access point. Some networks apply a hybrid model using a combination of continuous, event-driven and query-driven data delivery. We note that the framework we introduce here facilitates applications that are more related to the query-driven model. Many delivery protocols have been proposed for WSNs [25]-[30]. These protocols can be further classified into: hierarchy-based, data-centric or location-based [23]. Hierarchical protocols [25], [26] aim at clustering the nodes so that cluster heads aggregate and reduce the transmitted data in order to save energy. Data-centric protocols [27], [28] are query-based and depend on the naming of desired data, which helps in eliminating redundant transmissions. Location-based protocols [29], [30] utilize the position information to relay the data to the desired regions rather than broadcasting to the whole network, which reduces both bandwidth and energy consumption. We note that the framework we introduce here supports applications of a nature more related to data-centric protocols.
The routing protocol applied in a WSN is highly influenced by the data delivery model, especially with respect to managing the networks lifetime and energy resources. For instance, it has been shown in that for environmental monitoring applications where data is continuously transmitted to the access point, a hierarchical routing protocol is the most efficient alternative [25]. This is due to the fact that such an application generates significant redundant data that can be aggregated on route to the sink, thus reducing traffic and saving energy. Nonetheless, typical WSNs are too limited to fill the profile drawn by IoT in terms of heterogeneity and transmission/processing loads. Hence, we adopt an expanded notion of sensor networks that incorporates MANET nodes as well.

2.2.2 Placement in WSNs

Node placement is crucially important to achieve optimization goals in terms of coverage, connectivity or lifetime extension. WSN placement approaches may be classified into static and dynamic depending on whether the optimization is performed at the time of deployment or whiles the network is operational, respectively.

Static placement approaches may be further classified into controlled (grid-based) [63], [64], [65], [66] and random (Ad-hoc) [67], [69], [70]. Controlled node placement is necessary when sensors are expensive or when their operation is significantly affected by their position. Related scenarios include indoor applications, underwater applications and deployment of imaging sensors. Grid-based controlled deployment is a particularly attractive approach for coverage-oriented deployments due to its simplicity and scalability. Nonetheless, in practice, it is often infeasible to guarantee exact placement due to deployment errors such as misalignments and random misplacement [64]. Thus, for guaranteed coverage it is necessary to increase the grid resolution so that the deployment is resilient to these errors. In large-scale grid deployments,
however, relay nodes with advanced transmission capabilities must be placed to assure full connectivity which is proven to be NP-hard and requires applying optimization solutions [63], [71], [72]. We further note that the grid-based approach is mainly inapplicable for heterogeneous mobile settings and is not realistic for real-life scenarios. It is not cost effective in the sense that some grid points will have nodes but no phenomena to report.

In contrast to controlled placement, random sensor placement is performed in most WSN applications. This is due to the inaccessibility of the monitored areas [73]. Yet, random placement does not provide satisfactory coverage of the area unless an excessive number of nodes is deployed [28]. Moreover, Ad-hoc placement approaches entail the execution of redundancy-elimination algorithms to raise the cost-efficiency.

One option to improve coverage quality is to move nodes, such that sensors, relay nodes and even access points are relocated to areas where coverage or connectivity levels are unsatisfactory. The ability to dynamically reposition nodes while the network is operational is necessary to further improve its performance. Dynamic placement approaches for sensor nodes [19], [74], [75], [76], [77], [78] may be triggered by failure [74], poor coverage [75], [77], high traffic [76] or network partitioning [19]. The objective of node relocation among the aforementioned approaches varies from maintaining coverage [74], [75], [77] to lifetime [76] to connectivity [19], [78].

Dynamic relocation could be feasible in some applications such as environmental monitoring where sensors can be moved closer to the phenomenon to increase the fidelity of their data, or in military applications for the sake of keeping the base station out of the range of hostile fire. However, dynamic relocation does not apply to topologies where the sensor’s mobility follows human (e.g. cellular phones and medical sensors) or vehicular mobility traces. In fact, we
differentiate here between movable and mobile nodes. We state that the former may adhere to some dynamic placement schemes, while the latter are not necessarily susceptible to obey a relocation algorithm. The stochastic mobility nature of these nodes represents additional challenges to delivery and coverage solutions in wireless settings.

2.3 RFID-Sensor Networks

Depending on the intended application, integrating RFID and WSN components may follow several approaches including [47]:

1) Using RFID and sensors together to identify objects.
2) Using RFID tags for identifying objects while dedicating sensors for sensing.
3) Using RFID tags for identifying and WSNs for providing location.
4) Using RFID tags to assist positioning that is initially conducted by sensors.

Consequently, the literature recognizes three main combinations of RSN integration architectures: Tag-Sensor, Reader-Sensor and Mixed integration [14], [15], [79], [88]. We will refer to these main architectures in the remainder of this document as: TS, RS and MIX, respectively. All three architectures share some common components including: readers, tags, sensors and base stations. We next elaborate on each of these architectures.

2.3.1 Tag-Sensor Integration

Appending sensing capabilities to RFID tags is one of the simplest ways of integration [80]-[84]. The approach incorporates either providing RFID tags with sensing capabilities by equipping tags with sensory circuitry while restricting their communication to readers, or integrating tags with wireless sensor devices such that the integrated tags are able to communicate with other wireless devices in addition to readers. A wireless smart sensor platform is presented in [82]. It uses RF
links, such as WiFi, Bluetooth, in addition to RFID signals to communicate in a point-to-point fashion. In this architecture, a TS node uses the same protocols for reading tag IDs and for collecting sensed data. This option of integration limits the range of communication to RFID readers alone over single-hop links. In high-end applications, it is extremely desirable for integrated tag-sensors to communicate with each other as well as with other devices and form a cooperative Ad-hoc network as depicted in Figure 2.4.

It should be noted that a system that deploys a sensor with each tagged object is very costly and infeasible. The authors in [83] propose an approach that deploys an RFID tag attached to each sensor node. It provides both unicast and multicast capability. However, this integration architecture suffers from doubling the sensing load on each integrated node. The integrated entity is required here to run at least two wireless protocols depending on the sensed data and perhaps some aggregation method to overcome the short communication ranges of relaying sensors which increases the system’s operational and design costs especially under large-scale deployment.

Another TS approach is to integrate sensors with active tags [84]. This option implies using batteries to power the communication circuitry of the integrated element providing it with a longer range of communication. Nevertheless, because a battery is used, the cost and weight increases while its lifetime becomes limited. In an IoT deployment, the incorporation of sensors into tag designs is not realistic and defies the advantages tags provide in terms of size, weight and cost. Tags are usually attached to commercial goods and merchandise. It is highly impractical to include a sensor wherever a tag is located. Moreover, if the TS integration approach is to be delay-tolerant, then the size of the tag/sensor pairs would increase substantially to accommodate for the additional buffering circuitry.
2.3.2 Reader-Sensor Integration

The second architecture (RS) integrates RFID readers with sensor nodes [85], [86], [87]. Here, the existence of three types of devices is assumed: the integrated RFID reader-sensor nodes, simple RFID tags, and the base station (Figure 2.5). A prototype system is introduced in [87] for asset tracking with RFID and sensor networks. From a delay-tolerant perspective, providing integrated entities with extra buffering capacities may not present a design challenge since readers are already complex and large in size. However, depending on integrated RFID readers and sensors, to provide connectivity over the disrupted topology is not a cost-effective approach, especially when considering the limited sensing range and power consumption of sensors. As mentioned earlier, RFID readers are the most costly component of an integrated system. Considering an IoT setting where sensors are usually abundantly deployed, integrating readers with sensors will lead either to inflating the deployment costs due to the wide range of sensor distribution, or depriving sections of the topology from sensor coverage for the sake of reducing
the subsequent cost of integrated readers. Each of these alternatives has an effect on the system’s overall performance and efficiency.

**Figure 2.5. Integrated RFID reader-sensor (RS) architecture.**

### 2.3.3 Mixed Integration

In the mixed architecture (MIX) [14], [88], [89], RFID tags and sensor nodes coexist in the same network as distinct devices that are operating independently. Nevertheless, this necessitates designing a highly complicated integrated device to manage the two networks. The authors in [88] propose a MIX system composed of RFID networks, a WSN and an entity called an integration server which may initiate a task in the sensor network and may access the RFID network and assign tasks to it. In the MIX architecture proposed by [14], the system includes three classes of devices: sensor nodes, RFID tags and smart-stations (Figure 2.6).

A smart-station consists of an RFID reader combined with a data microprocessor and a network interface communicating exclusively with the network’s base station. This approach is vulnerable to large-scale implementations, such as in IoT settings, where data relaying is highly required. It
also presents some problems related to energy imbalance among the smart nodes it proposes. The authors in [89] note that in the architecture of [14], smart nodes have a fixed transmission range. Hence, the amount of traffic that is to be forwarded increases considerably as the distance to the base station becomes shorter. Subsequently, smart nodes that are closer to the base station will run out of power earlier causing partitioning. We further add that the integration approach in [14] does not address neither the placement nor the cost associated with the reader component of the proposed smart-stations. In general, we consider the three integration architectures mentioned above to be not suitable for IoT environments since none of them consider ambient wireless devices to tackle connectivity challenges. Moreover and most importantly, none of the three aforementioned architectures address the concept of delay-tolerance, which is concurrent to data delivery in IoT.

![Diagram of Mixed Integration (MIX) architecture of RFID and WSN.](image-url)

**Figure 2.6. Mixed integration (MIX) architecture of RFID and WSN.**
2.4 Summary

RSN integration merges the capabilities of the two parenting technologies in various forms, depending on the implemented integration architecture. The three main RSN integration approaches are summarized in Table 2.2. However, none of the surveyed integration architectures fully address delivery constraints from an IoT perspective, particularly in terms of nodal placement and delay-tolerance. Placement was discussed from two perspectives. From the RFID perspective, reader placement is specifically important in order to achieve maximum area coverage and to avoid reader collisions. On the other hand, sensor placement optimization schemes are classified depending on whether the optimization is performed at the time of deployment or while the network is operational. The latter suggests that the nodes have a level of mobility, which is not the common case in WSN. In order to incorporate mobility and traffic types associated with IoT, we expanded our notion of sensors to include MANET nodes and ambient wireless devices in general.

Table 2.2. Summary of RSN integration architectures.

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Properties</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag-Sensor (TS)</td>
<td>Sensor-tags can intercommunicate with other devices. Sensors are traceable by RFID addresses.</td>
<td>Integrated nodes not as convenient as tags in terms of size and lifetime. Massive deployment of sensors add to the system’s cost</td>
</tr>
<tr>
<td>[80], [81], [82], [83], [84]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reader-Sensor (RS)</td>
<td>Sensor-reader intercommunication range exceeds the interrogation range of sole readers.</td>
<td>Adding a reader to each sensor in a large-scale topology is infeasible and not cost-efficient solution. It also introduces serious placement challenges. In terms of both sensor and readers coverage.</td>
</tr>
<tr>
<td>[85], [86], [87]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mixed integration (MIX)</td>
<td>Sensors and tags exist separately in the system.</td>
<td>There is a need to introduce complex integrated hardware entities to organize the tasks of the two networks.</td>
</tr>
<tr>
<td>[14], [88], [89]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Chapter 3

Optimized Placement in RSNs

Integrating RFID and WSN technologies has many challenges and constraints. The most obvious refer to the additional costs related to designing integrated hardware components. Such components are naturally more expensive than simple sensors or tags and may operate both wireless communication protocols either simultaneously or alternatively. This puts additional operational load on the integrated system which translates into additional energy consumption. Thus, precise and non-redundant placement of integrated devices is a critical factor in determining the cost efficiency of any integrated RSN system.

We approach the placement problem in RSN topologies by first introducing a novel integration approach that concentrates the cost factor in a single RSN component composed of a relay and an RFID reader. We define this integrated node as a Super Node (SN). Based on this approach, minimizing the system’s cost implies minimizing the count of SNs in the topology, which consequently requires their optimal placement to be in a manner that attains maximum coverage. We use Integer Linear Programming (ILP) to achieve optimal placement. The remainder of this chapter is organized as follows. Section 3.1 outlines the proposed approach and highlights its contributions. Section 3.2 presents the system models including the integration architecture, its deployment cost analysis and its communication model. Section 3.3 describes our ILP-based placement strategy for the proposed integration approach. Section 3.4 presents the simulated cost evaluation of our approach against other integration architectures. Lastly, Section 3.5 concludes this chapter.
3.1 Outlines and Contributions
We propose a novel RSN architectural model that is dependent on the ability of Super Nodes (SNs) to aggregate data from both sensors and RFID tags and then transmit it to base stations. Optimal placement of SNs is critical for the efficiency of our architecture due to a number of considerations. First, achieving coverage of both sensors and tags demands efficient controlled, as opposed to random, placement of SNs across the topology. Second, the presence of the reader component in SNs necessitates a placement scheme that avoids reader-to-reader collisions. Lastly, redundancy in SNs is to be prevented as well, in order to achieve cost-efficiency since SNs are the most complicated and costly component of our model in terms of hardware design and energy consumption. To this end, the contribution of this chapter is twofold:

1) We introduce a novel architectural approach for Smart Integrated WSNs and RFIDs (SIWR). Our approach combines readers and relays as SNs. The functional and cost efficiencies of this architecture, however, are dependent on the accurate placement of SNs.

2) We provide the ILP formulation for optimal placement of the minimal number of SNs while guaranteeing main connectivity and coverage constraints. The results of our experiments show that our proposed architecture outperforms other common integration architectures in terms of cost-effectiveness.

3.2 System Models
In the section, we elaborate on our network model. This includes a detailed description of the components of our novel architecture, in addition to comparing its deployment cost against other common RSN integration approaches. We provide a description of the communication model of the architecture that we base our placement scheme on.
3.2.1 Network Model

We call our integration architecture SIWR: Smart Integrated WSNs and RFIDs. It is represented by a three-layer hierarchy that maximizes the integrated network usability and minimizes its deployment cost. The components of our proposed SIWR architecture are: Light Nodes (LNs), Super Nodes (SNs) and Base Stations (BSs). In the following, we describe the functionality and role of each of the components of SIWR.

**Light Nodes**

Light Nodes (LNs) represent the data sources residing at the lowest tier of SIWR. Any data exchanged across our RSN architecture is assumed to have been originated at the LN level. LNs consist of simple sensors and tags that primarily exist as separate nodes. According to our model, LNs are fully dedicated to performing data gathering/reporting and are relieved from conducting any relaying or processing tasks. This arrangement contributes to prolonging their operational lifetime. We note that LNs’ deployment cost is relatively low and, hence, has a minor impact on the network’s cost.

**Super Nodes**

We introduce Super Nodes (SNs) as a novel and unique RSN component in terms of both integration approach and functionality. SNs are made of integrated wireless relaying and RFID reading components. Hence, SNs perform a dual-functionality with respect to both WSNs and RFID systems. First, SNs are equipped with advanced processing and transmission units to aggregate and relay data from the sensors among LNs. Second, SNs act as RFID readers of data stored on or provided by tags within their interrogation vicinities. SNs are capable of inter-communicating their aggregate tag/sensor data, as well. The SIWR architecture aims toward dominating the cost factor by distributing the sensing and relaying loads over the components of
the integrated networks in an optimum fashion. The functional and design complexity of SNs imposes the need for their efficient deployment in order to control the cost factor in addition to guaranteeing coverage and avoiding interference.

**Base Stations**

A Base Station (BS) acts as a sink for the data aggregated by SNs. A BS is directly connected to the Internet. In SIWR, BSs are assumed to be owned by independent bodies. Hence, their operation and deployment is not controlled by our approach.

Figure 3.1 depicts the SIWR network model with the aforementioned components in a 3-D topology of several LNs and four SNs (SN1 - SN4) communicating with the same BS. In this topology, LNs (i.e. simple sensors and tags) are assumed to be placed near to the phenomenon of interest. A SN is supposed to be placed on the most appropriate position to serve the largest number of LNs distributed around it based on the application’s requirements. However, Figure 3.1 illustrates a situation where among the four SNs, only two (SN1 and SN2) are properly placed. SN3 is redundant and interferes with SN2. On the other hand, SN4 suffers from a misplacement error and does not cover any of its intended LNs.
3.2.2 Deployment Cost

System deployment cost is estimated with respect to the main components of the integrated architecture. In order to set a baseline for this cost analysis, we recall the three main RSN integration approaches discussed in Section 2.3: TS, RS and MIX. The common components of these architectures are readers, tags, sensors and relay nodes. However, the cost of tags and sensors are relatively marginal with respect to that of readers and relays. RFID readers employ
complicated and expensive circuitry. The same could be said about relays compared to sensors. Hence, we will refer to the functionalities of these components to quantify the cost measure of each integrated architecture as follows:

1) $Cost_{Read}$: Cost of reader nodes.
2) $Cost_{Relay}$: Cost of relaying nodes.
3) $Cost_{XS}$: Cost of extra sensor nodes that may be deployed for relaying and sensing.

Thus, if we define the cost of TS integration architecture, it would be dominated by the total count of readers and the extra sensors used for relaying:

$$Cost_{TS} = Cost_{Read} + Cost_{XS}$$ (3.1)

As for the RS integration, we note that combining readers with sensors in one entity will lead to either inflating the deployment costs due to the sensors’ wide distribution range or to depriving wide sections of the topology from coverage for the sake of reducing the subsequent cost of integrated readers. Each of these alternatives has its effect on the system’s overall deployment cost measure, which is dominated by the same factors as its previous:

$$Cost_{RS} = Cost_{Read} + Cost_{XS}$$ (3.2)

In the MIX integration, sensors and tags exist as independent entities and their cost impact is relatively marginal. Here, readers and smart-stations are the most complex components whose deploying will substantially affect the system’s cost. Hence, the cost of this mixed architecture is more complicated and can be expressed by the sum:

$$Cost_{MIX} = Cost_{Read} + Cost_{Relay} + Cost_{XS}$$ (3.3)

Our SIWR integration architecture, on the other hand, dominates the cost factor by focusing reading and relaying loads in one integrated component (SNs). This approach is unique because it
does not overwhelm the light sensor and tag nodes (LNs) with relaying tasks but rather focuses on the optimal distribution of the most costly SNs. When compared to the aforementioned integrated architectures, the cost measure of SIWR is expressed by the total count of SNs, where a single super node cost is calculated by:

\[
Cost_{SN} = \varepsilon Cost_{Relay} + \eta Cost_{Read},
\]

where \(\varepsilon\) and \(\eta\) denote fractional variables varying between 0 and 1 based on the hardware specifications of the designed SN. We note that Eq. 3.4 combines partial costs of readers and relaying nodes which are conducted separately in the other architectures. This is due to the elimination of duplicated components achieved by our integration strategy.

### 3.2.3 Communication Model

We assume a probabilistic model in which the probability of communication between two wireless devices decays exponentially with distance and takes into consideration surrounding obstacles. Accordingly, the communication range of each device can be represented by a 3-D arbitrary shape. For realistic estimation of the arbitrary shape dimensions, we need a practical signal propagation model. This model can describe the path loss (the difference between transmitted and received signal power) in the targeted site by taking into consideration the effects of the surrounding terrain on the power \((P_r)\) of received signals as follows [90]

\[
P_r = K_0 - 10\gamma \log(d) - \mu d,
\]

where \(d\) is the Euclidian distance between the transmitter and receiver, \(\gamma\) is the path loss exponent calculated based on experimental data, \(\mu\) is the mean of normally distributed function describing signal attenuation caused by shadowing and multipath effects in the monitored site, and \(K_0\) is a constant calculated based on the transmitter, receiver and field mean heights.
Let $P_c$ equal the minimal acceptable signal level to maintain connectivity. Assume $\gamma$ and $K_0$ in Eq. 3.5 are also known for the specific site to be monitored. Thus, a probabilistic communication model $P_c$ for the probability that two devices can communicate with each other is given by

$$P_c(d, \mu) = Ke^{-\mu\gamma},$$ (3.6)

where $K_0 = 10\log(K)$.

The probabilistic connectivity $P_c$ is not only a function of the distance separating the sensor nodes but also a function of the surrounding obstacles and terrain, which can cause shadowing and multipath effects (represented by the random variable $\mu$).

### 3.3 Placement Strategy

The SN placement problem proposed in this chapter has infinitely large search space and finding the optimal solution is of the utmost importance. Therefore, we propose a 3-D grid model that limits the search space to a more manageable size. We assume knowledge of the 3-D terrain of the monitored site. Hence, practical candidate positions on the grid vertices can be pre-determined. Non-feasible positions are excluded from the search space.

We use the aforementioned cubic grid’s vertices to apply a novel placement scheme for SNs in an integrated SIWR architecture. This strategy is used to minimize the cost of the integrated network without violating the main requirements of RFID and WSNs. The former requires maintaining the right ratio of tag to reader counts, while the latter requires full connectivity. Our placement scheme aims at solving the following problem:

*Find the optimal locations of the least $SN_{\text{total}}$ super nodes with the routing paths to deliver the generated data from each tag/sensor to the base station.*
The optimization problem can be formulated as an ILP. We define the following constants and variables:

**Constants:**

\[ V \] is the set of candidate grid vertices.

\[ v \] is the number of candidate positions on the grid vertices.

\[ SN_{total} \] is the total available super nodes.

\[ f_{ij} \] is the flow from node \( i \) to node \( j \) (i.e. the data units to be sent from \( i \) to \( j \)).

\[ G_i \] is the generated traffic by sensor node \( i \).

\[ SG_i \] is the generated traffic by super node \( i \).

\[ C_i \] is the capacity of traffic (BW) available for sensor node \( i \).

\[ SC_i \] is the capacity of traffic (BW) available for super node \( i \).

**Variables:**

\[ \alpha_i \] is a binary variable equals to 1 when a sensor is placed at vertex \( i \) of the 3-D grid and 0 otherwise.

\[ \beta_i \] is a binary variable equals to 1 when an RFID tag is placed at vertex \( i \) of the 3-D grid and 0 otherwise.

\[ S_i \] is a binary variable equals to 1 when an SN is placed at vertex \( i \) of the 3-D grid and 0 otherwise.

\[ P_c(i, j) \] is the probabilistic connectivity between two nodes placed at vertices \( i \) and \( j \).
is a set of neighboring indices such that \( j \in N(i) \) if node \( j \) is within the transmission range of node \( i \) (i.e. \( P_r(i, j) \geq \tau \)), where \( \tau \) is the connectivity threshold.

\( M(N(i)) \) is a set of indices such that \( j \in M(N(i)) \) if node \( j \) is within the transmission range of a node that can reach one of the neighboring nodes of node \( i \) via single or multiple hops.

\( T_{ij} \) is a binary variable equals to 1 if SN \( i \) is transmitting to SN \( j \) and 0 otherwise.

Our goal to minimizing the cost implies minimizing the total count of SNs without affecting the main connectivity requirements of WSN and RFID systems. By achieving a placement that maintains the maximum number of RFID tags per reader, we optimally handle the main connectivity requirement in the placement of RFID systems. On the other hand, we assure that each sensor will be connected to the base station through at least one path. In order to do so, we formulate the ILP in Table 3.1.

Eq. (3.7) is the objective function which minimizes \( SN_{total} \). Eqs. 3.8 and 3.9 ensure that each sensor node is connected to at least one super node, and each tag has at least one super node in its vicinity, respectively. Eq. 3.10 ensures at least one path towards the BS from an SN which is not a neighbor (i.e. have direct communication) with the BS. Eq. 3.11 guarantees that the total number of tags covered by a super node (reader) is not exceeding the Optimized Value (OV) of tags per reader [62]. Eqs. 3.12 and 3.13 satisfy the traffic capacity (bandwidth) constraints. We note that this ILP can be easily modified to handle more complex capacity constraints (by giving different weights for different links of a single node). Eqs. 3.14 and 3.15 guarantee the flow balance. Eqs. 3.11 - 3.15 together are used to ensure the least interference in the wireless medium access. Eq. 3.16 prevents the flow splitting by specifying that a super node \( j \) can transmit to only one super node \( i \).
Table 3.1. ILP formulation for SIWR.

\[
\begin{align*}
\text{Minimize} & \quad SN_{\text{total}} & (3.7) \\
\text{Subject to} & \\
\sum_{i=1}^{v} S_j. \alpha_i & \geq 1, \quad \forall j \in V & & j \in N(i) \quad (3.8) \\
\sum_{i=1}^{v} S_j. \beta_i & \geq 1, \quad \forall j \in V & & j \in N(i) \quad (3.9) \\
S_j. (\sum_{j \in N(i) \& j \in N(\text{BS})} S_i) & \geq 1, \quad \forall j \not\in \text{BS} \quad (3.10) \\
\sum_{i=1}^{v} S_j. \beta_i & \leq OV, \quad \forall j \in V & & j \in N(i) \quad (3.11) \\
\sum_{j \in N(i)} \alpha_i. f_{ij} & \leq C_i, \quad \forall i \in V \quad (3.12) \\
\sum_{j \in N(i)} S_i. f_{ij} & \leq SC_i, \quad \forall i \in V \quad (3.13) \\
\sum_{j \in N(i)} \alpha_i. f_{ij} - \sum_{k \in N(i)} \alpha_i. f_{ki} & = G_i, \quad \forall i \in V \quad (3.14) \\
\sum_{j \in N(i)} S_i. f_{ij} - \sum_{k \in N(i)} S_i. f_{ki} & = SG_i, \quad \forall i \in V \quad (3.15) \\
\sum_{i=1}^{v} T_{ji} & = S_j, \quad \forall j \in SNs \quad (3.16)
\end{align*}
\]
3.4 Experimental Results

In this section, we evaluate our SIWR architecture against the three common RSN integration architectures discussed earlier. Namely: TS, RS and MIX. We apply the introduced ILP-based deployment strategy on a 500×500 m graph while increasing the density of SNs from 3, to 30, to 300 SNs/m², as shown in Figures 3.2, 3.3 and 3.4, respectively.

Each of the LN densities is applied over the four integration architectures simultaneously and the resulted count of SNs necessary to satisfy the connectively requirements in each of the four architectures is plotted against the corresponding LN count. We remark that the SNs count in case of TS, RS, and MIX architectures will be the count of total readers and sensors relaying data to the base station. The different architectures are evaluated and compared using two metrics: cost and SN count. The cost metric is represented by an increase in a monetary value with respect to the total count of relays, readers, and/or SNs according to the cost functions mentioned Section 3.2.2. The SN count reflects the scalability of the utilized architecture under varying application scales, in addition to reflecting the application’s complexity.

The proposed ILP formulation is solved using MATLAB. We use MATLAB lp-solver v5.5 with a timeout of minutes. In other words, the ILP of a particular round is solved during the last minutes of the previous round. Based on experimental measurements [90], we set the communication model variables (defined in Section 3.2.3) to be as follows: $\gamma = 4.8$, $P_r = -104$ (dB), $K_0 = 42.152$, and $\mu$ to be a random variable that follows a log-normal distribution function with mean 3 and variance of 10. We choose $\epsilon = \eta = 0.8$ according to [91]. The MATLAB simulator determines whether a wireless node is connected to its neighbors or not based on the aforementioned probabilistic communication model, where $\tau = 70\%$. Each simulation experiment
is repeated 100 times and the average results hold a confidence interval no more than 5% of the average (over 100 runs) at a 95% confidence level.

The results as plotted in Figures 3.2 – 3.4 show that our SIWR architecture, indeed, outperforms all the three common RSN integration architectures and always requires less number of SNs. This translates into minimal cost for an optimal deployment.

In our first ILP iteration, we assumed a density of 3 tags/sensors per m². As shown in Figure 3.2, the count of SNs increase almost linearly for the four architectures as the count of LNs increase, as to be expected naturally. Yet, the SIWR architecture maintained the least number of SNs throughout the iteration.

![Graph showing cost comparison of different architectures with 3 SNs/m².](image)

**Figure 3.2. SIWR cost comparison with 3 SNs/m².**
The same observation, with even better performance (i.e. fewer SNs required by SIWR) holds true when increasing LNs density to 30/m², as shown in Figure 3.3. However, when this density is increased to 300 LNs/ m² and the total count of tags/sensors in the topology reaches the range of thousands, which is trivial in many large-scale applications involving WSNs and RFIDs, the performance differences between SIWR and the competing integration architectures becomes astonishingly obvious.

Figure 3.3. SIWR cost comparison with 30 SNs/m².
As shown in Figure 3.4, when the count of LNs exceeds 2000, the corresponding count of SNs to connect the set of tags/sensors increases exponentially and so does the deployment cost of the involved architectures. This is not the case with SIWR which maintains a steadier rate of SN count, which remains lower than counts required by any of the other architectures. This advantage is maintained as the number of LNs increases (i.e. deployment area increases) and strongly implies that SIWR represents the best integration architecture for RSN applications in terms of deployment and energy consumption costs.

Figure 3.4. SIWR cost comparison with 300 SNs/m².
3.5 Summary

In this chapter, we proposed a novel cost-effective hierarchical architecture for Smart Integrated WSNs and RFIDs (SIWR). Our integration approach is based on minimizing deployment costs by optimally determining the placement of the most expensive elements of the integrated system, which are represented in SIWR by Super Nodes (SNs). SNs are composed of integrated relay and RFID reader pairs. By choosing to integrate these two particular components, we aim towards balancing the distribution of relaying and data collection evenly over our system. In SIWR, SNs aggregate and transmit data from the lower tier, represented by simple sensors and tags or Light Nodes (LNs), to the system’s Base Stations (BSs).

The placement of SNs is of utmost importance due to its significance in terms of performance and cost-efficiency. We introduce an Integer Linear Programming (ILP) based placement strategy to determine the optimal locations for the minimum number of SNs. When compared against other integrated architectures, simulation results showed that our SIWR approach and ILP-based placement strategy outperform all the competing architectures even over higher densities of nodes. The deployment cost required by SIWR showed to be definitely less than those of other architectures for the same numbers of sensor nodes and tags in any given integrated layout.
Chapter 4
Data Delivery

Data delivery in RSN architectures under the umbrella of IoT has to adhere to numerous metrics such as delay-tolerance, connectivity and lifetime. These metrics contribute to the level of efficiency and performance of proposed delivery schemes according to the design parameters which may include scalability, delivery rate and quality of service. In this chapter, we build on SIWR to present a modified RSN-based architecture that further recognizes ubiquitous mobile nodes in IoT and employs them as Courier Nodes (CNs). Consequently, we present two data delivery schemes that address delay and connectivity objectives in RSN architectures, respectively.

The first delivery scheme we introduce is (DIRSN); an optimized Delay-tolerant approach for Integrated RFID-Sensor Networks. This is a delay-based novel scheme for data delivery and CN selection in RSNs. DIRSN considers the variations between nodes in IoT in terms of mobility and connectivity capacities. By associating these variations to the fact that an IoT setting is frequently disrupted, we employ CNs into a new decentralized ILP-based delay-tolerant approach that locates the optimum set of CNs per time-round. This approach aims toward guaranteeing minimum-delay-connectivity between Super Nodes (SNs) and Base Stations (BSs) by locating the best set of CNs capable of doing so without violating the main requirements of RFIDs and WSNs that are imposed by the dense deployment and node/link capacities such as load balancing.

The second data delivery approach we propose is (URIA); a Ubiquitous Robust Integrated Approach. This connectivity-based approach deploys CNs as linkers between SNs and BSs in an RSN in order to guarantee a specific level of connectivity specifically for deployments in harsh
environments. In addition, URIA maintains constraints on delay, such that a multi-path minimal-delay route is always provided between any source-destination pair. Our approach is formulated via a Semi-Definite Programming (SDP) solution and is compared against other RSN integrated schemes targeting connectivity and delay metrics.

The remainder of this chapter is organized as follows. Section 4.1 outlines the proposed approaches and highlights their contributions. Section 4.2 describes our delay-based data delivery scheme (DIRSN) in terms of its system models, methodology and experimental results. Section 4.3 describes our connectivity-based data delivery scheme (URIA) in terms of its system models, methodology and experimental results. Section 4.4 concludes this chapter.

4.1 Outlines and Contributions

In this chapter, we introduce two data delivery schemes for integrated RSN architectures. The first scheme (DIRSN) is delay-based, meaning it satisfies delay-tolerance constraints by employing courier nodes, in addition to executing an ILP-based solution to select couriers with minimal delay connections. The second scheme (URIA) is connectivity-based. URIA executes an SDP-based solution that determines the optimum subset of couriers in a manner that insures specific connectivity level with minimal delay across the network.

To this end, we list the contributions of this chapter as follows:

1) We introduce Courier Nodes (CNs) to our RSN integration approach as ubiquitous linkers between the previously defined SNs and BSs to facilitate our delay and connectivity based delivery approaches.

2) We introduce a delay-based delivery scheme called DIRSN (Delay-tolerant approach for Integrated RFID-Sensor Networks). DIRSN employs a novel ILP-based formulation that
guarantees best CN selection to minimize the delay across the integrated topology while obeying link-capacity and load balancing constraints.

3) We introduce a connectivity-based delivery approach for RSNs called URIA (Ubiquitous Robust Integrated Approach). URIA utilizes the mobility of CNs to guarantee a specific level of connectivity between SNs and BSs. It incorporates an SDP-based solution to achieve a guaranteed connectivity level, in addition to catering to delay constraints for applications that are not delay-tolerant.

4.2 Delay-Based Delivery Scheme

Delay is considered to be a main delivery challenge in IoT settings. In fact, a research discipline related to a class of wireless networks, known as Delay-Tolerant Networks (DTNs), is motivated by common and long lasting partitions experienced due to sparse distribution of nodes in a given topology [21]. A DTN may be characterized by any combination of the following:

1) Intermittent connectivity: If there is no consistent end-to-end path between the source and destination; a phenomenon known as network partitioning.

2) Ambiguous mobility patterns: Unlike the case with public bus services that maintain fixed routes or planetary trajectories, future behavior of a node is not fully known.

3) Long or variable delays: Long propagation delays between nodes, in addition to variable queuing delays at node buffers, all create end-to-end delays that far exceed the threshold levels usually tolerated by Internet protocols and applications that rely on quick return of acknowledgements.

We have defined delay-tolerance as a main characteristic of IoT setting. Accordingly we define delay-tolerant data.
Definition 1. Delay-Tolerant Data (DTD): is data that may be delivered to its ultimate destination within an extended time-to-live $\Delta$, where $\Delta$ is application-dependent, without losing its validity.

Data delivery in DTNs is substantially different than delivery in traditional WSNs or MANETs. Ad-hoc protocols such as Optimized Link State Routing (OLSR) [31], Ad-hoc On demand Distance Vector (AODV) [33] and Dynamic Source Routing (DSR) [34] implicitly assume that the network is connected and there is a contemporaneous end-to-end path between any source/destination pair. Under these protocols, when packets arrive and no end-to-end paths for their destinations can be found immediately, these packets are simply dropped. To achieve delivery and cope with intermittent connectivity constraints, DTNs perform a Store-Carry-Forward (SCF) [22] approach that demands buffering the data at the intermediate node until a next forwarding opportunity arises.

Extensive research has been done on data delivery in DTNs [93]-[102]. The selection of the most appropriate delivery protocol depends on the level of knowledge available regarding the networks’ topology. Such knowledge may be absolutely absent and necessitates blind [93], [94], or partial flooding [95], [96] or based on update table exchange among neighboring nodes [97], [98], [99]. A class of DTN delivery schemes exploits the mobility of nodes [100], [101], [102] either by proactively modifying the trajectory of some nodes [102] or simply relaying on the randomness of the mobility pattern [100] in a way that enhances the system’s performance (i.e. reduces the transmission delay of messages). In message ferrying [102], special mobile nodes with deterministic mobility and limited storage capacity periodically transit from one location to another carrying data between other disconnected stationary nodes in the network. The scheme presented in [100], on the other hand, utilizes nodes
(DataMULEs) with arbitrary mobility patterns. The proposal consists of a three-tier architecture (sensor nodes, DataMULEs and access points) to connect spare sensors at the cost of high latency. In the DataMULEs architecture, sensor nodes are dedicated to sensing while DataMULEs initiate communication with sensors, collect their data and deliver it to access points. This arrangement aims at minimizing energy consumption with respect to a sensors’ lifetime. We propose a delay-tolerant delivery approach that is, on the one hand, history-based and encourages the exchange of routing tables between neighboring nodes. On the other hand, we employ the mobility of pervasive or ubiquitous nodes in IoT which act similar to DataMULEs or message farriers, depending on the level of randomness governing their mobility patterns. We argue that this approach is generally applicable to IoT settings due to the widespread of wireless smart devices which we utilize as CNs. To this end, we introduce DIRSN as a novel scheme for delay-based data delivery in integrated RSNs. In the remainder of this section, we present DIRSN’s system models and its courier selection strategy, in addition to its experimental results.

4.2.1 System Models
We describe in the following the network component of the RSN architecture upon which we run our DIRSN scheme. We also introduce our delay model for our delay-based ILP formulation.

4.2.1.1 Network Model
We build upon the three-tier hierarchy previously presented in Section 3.2.1 and the probabilistic communication model presented in Section 3.2.3. Henceforth, our data delivery scheme is designed with respect to the following components:

1) Base Stations (BSs) that are fixed and directly connected to the Internet cloud acting as sink nodes for the rest of the nodes down the hierarchy.
2) Super Nodes (SNs) representing the integrated part of our architecture. SNs perform the roles of RFID readers and wireless relays to BSs, simultaneously, with advanced transmission capabilities. Minimizing the number of these SNs will minimize the deployment cost of the system as a whole since they are the most sophisticated and expensive among its components.

3) Light Nodes (LNs) represented by passive RFID tags and simple sensor nodes, each performing their own protocol. LNs are assumed to be distributed densely over the topology and may be fixed or mobile depending on the application.

4) In addition to the aforementioned components, the DIRSN architecture recognizes pervasive mobile nodes naturally available within the IoT realm as Courier Nodes (CNs). CNs are represented in our model by smart devices and vehicles equipped with transceivers, variable buffering and communicational capabilities. A CN is potentially moving towards or residing within the communication range of a BS. Whenever connectivity is lost between SNs and BSs, a CN may act as a linker conducting SCF (i.e. buffering the SN’s data) until a suitable best-next-hop is found or the BS is eventually reached. If its transmission power permits, a CN may even assume the role of a relay and directly transfer the SN’s load to a BS.

Figure 4.1 illustrates the roles of the four components of the DIRSN architecture where CNs are represented by smart devices and transceivers onboard vehicles. Figure 4.2 depicts a number of scenarios demonstrating the capabilities of each of the DIRSN architecture components. Note that in Figure 4.2, there are three fixed SNs: \(X\), \(Y\) and \(Z\). There are also three mobile CNs: \(A\), \(B\) and \(C\), in addition to several LNs. In this setting, the three SNs are sufficiently placed according to SIWR (See Chapter 3) to provide full coverage to any LN in the topology. Furthermore, we assume that SNs \(X\) and \(Y\) are incapable of directly communicating.
with any BS for power constraints. As for the CNs, we see that node $A$ has a mobility pattern that allows it to gather data from both SNs $X$ and $Y$. Upon reaching SN $Y$, which is the end of $A$’s path, node $A$ realizes that it is still not in the range of any BS, so it decides to transfer its load a hop further to a neighboring courier $C$ that shows a potential (i.e. by examining the mobility history of $C$) of contacting a BS in the future. $C$ also reads from SN $Y$, which simultaneously forwards another set of delay-intolerant packets to CN $B$, whose routing table indicates a better probability of communicating with a BS sooner. Meanwhile, $C$’s mobility allows it to finally deliver its data load to a BS. Note that SN $Z$ was able to reach a BS by itself regardless of the assistance offered by any CN. Hence, it did not forward its packets to $C$ when it was in its proximity.

Figure 4.1. Network model of DIRSN.
Figure 4.2. Connectivity role of CNs according to DIRSN scheme.

4.2.1.2 Delay Model

Our objective in DIRSN is to minimize the worst delay experienced between any SN-BS pair. We adopt the delay model introduced in [103]. Although that model was initially proposed for underwater acoustic sensor networks, we find it generally suitable for wireless networks regardless of the transmission medium. Most importantly, since we are using Integer Linear
Programming (ILP), we assume its discretized delay metric that can be tuned to achieve any desired accuracy.

Due to dense network topologies formulated in IoT, a relatively long multi-hop path can easily exist between the source node and the corresponding BS. The delay components we consider in this chapter are the transmission/processing delay $\psi$ modeled by the number of hops multiplied by $\psi$, and the propagation delay, modeled based on the speed of signal and the Euclidian distance between two ends. The latter delay varies based on the utilized technology and its corresponding standards, and the transmission medium. Accordingly, we define a delay step $\omega$ which is the distance a wireless signal would travel in one time unit. Let $E_{ij}$ be the Euclidian distance between a source node $i$ and a destination node $j$, then, the discrete propagation delay over a single-hop link $(i, j)$ would be $\left\lfloor \frac{E_{ij}}{\omega} \right\rfloor$. Hence, the discrete delay over a multi-hop path is the sum of the discrete delays of single-hop links that constitute that path. Note that $\omega$ (and the time unit) can be made small enough to meet any desired accuracy. Single-hop-delay ($D_{single}$) and Total-delay ($D_{total}$), can be respectively defined as follows:

$$D_{single} = \left\lfloor \frac{E_{ij}}{\omega} + \psi \right\rfloor$$

(4.1)

and

$$D_{total} = \sum_{total\ hops} \left\lfloor \frac{E_{ij}}{\omega} + \psi \right\rfloor$$

(4.2)

4.2.2 Problem Definition and Formulation

DIRSN aims at minimizing the delay in the integrated IoT network without violating the main requirements of RFIDs and WSNs. DIRSN, aims at solving the following problem:
Given a large search space (modeled by IoT random deployment) and a limited number of SNs and static/mobile LNs and CNs, Find the optimal set of CNs with their corresponding routing paths to deliver the generated data from each SN covering tag/sensor LNs to the BS in minimum delay based on current network topology.

We assume discretized rounds per which our scheme selects routing paths based on locating the most suitable couriers passing through these paths. This is determined according to minimum delay and link constraints calculated from the lower layer of the architecture (LNs) up to SNs. Figure 4.3 shows an example where four SNs: A, B, C and D, respectively, have established at some point of time one or more connecting paths, through some CNs, to one BS. Note that each SN-BS path has its end-to-end delay and capacity characteristics given in the format (delay/capacity). These characteristics are based on table exchanges between CNs and SNs. The ILP of DIRSN selects the path (illustrated by the thick red line) that guarantees minimum delay without violating capacity constraints. The decision to utilize this CN depends on its resources which are stored, as well, at the SN’s routing table. For example, assuming that SN A in Figure 4.3 has to transmit packets with a load-balance constraint that requires a minimum link capacity of 10MB, the ILP detects three different paths connecting A to the BS. Two of the three paths obey the capacity constraints, and of these two the path which provides the minimum delay is ultimately chosen. Note that the third path does provide minimal delay from A to the BS. However, it fails to fulfill the required capacity hence it was not chosen by the ILP. The case is easier with SN B, where there is only a single path connecting it to the BS. The ILP is forced to select this path, regardless of its promised delay once it fulfills the capacity constraint. As for SN C, there are two paths with equal link capacities linking it to the BS, one of which passes through
another SN $D$. Obviously, the path that provides minimum delay is selected by the ILP in this case.

![Diagram of COURIER selection](image)

**Figure 4.3. Example of DIRSN courier selection.**

The selection of the best $M$ couriers in the proximity of each SN and available CN in the network is achieved by periodically exchanging routing tables and/or registration records between neighboring nodes. This selection process is repeated at the beginning of each triggered round. The mobility history of the neighbors is examined against the communication range of the corresponding destination node(s). Based on the results, forwarding candidates are defined according to best (i.e. minimum) delays. The algorithm we use to conduct this approach is based on an ILP formulation which, in turn, requires defining the following constants and variables.
**Constants:**

- $C_{total}$ is the total available couriers for data transfer.
- $D_{max}$ is the maximum delay to transfer a unit of data from a SN to the courier.
- $SG_i$ is a data generation rate of a super node $i$ (based on the underlining connected LNs per SN).
- $t_i$ is the traffic capacity of a super node $i$ (i.e. maximum data units that can be relayed by a SN per round).
- $T_i$ is the traffic capacity of a courier node $i$ (i.e. maximum data units that can be relayed by a courier per round).
- $E_{ij}$ is the Euclidian distance between node $i$ and $j$.
- $\omega$ is the distance a node signal would travel in one time unit.
- $M$ is the candidate couriers’ count available per round.
- $N(i)$ is a set of neighboring candidates such that $j \in N(i)$ if node $j$ is within the transmission range of node $i$.
- $M(i)$ is a set of indices such that $j \in M(i)$ if node $j$ is within the range of a CN $i$ that can reach an BS.

**Variables:**

- $c_i$ is a binary variable equals to 1 when a CN at position $i$ (associated with an $(x,y)$ coordinate) is chosen by an SN to relay its data to the BS, and 0 otherwise.
- $f_{ij}$ is the flow from SN $i$ to CN $j$ (i.e. the data units to be sent from $i$ to $j$).
- $l_{ij}$ is the flow from a CN $i$ to CN $j$.

Minimizing delay implies minimizing the total path length towards the BS. This is achieved by locating a courier set that maintains the shortest path from each SN to a BS while considering
their varying node/link capacities and load balance. In order to do so, we formulate the ILP in Table 4.1. Eq. 4.3 is the objective function which minimizes $D_{\text{max}}$. Eqs. 4.4 and 4.5 satisfy the traffic capacity constraints available to SNs and CNs, respectively. Eq. 4.6 guarantees the flow balance and controls data generation rates at SNs. Eq. 4.7 guarantees that if no courier is selected (i.e. $C_j = 0$), no flow is sent to courier at position $j$. Eq. 4.8 sets $D_{\text{max}}$ as the maximum delay over all SNs seeking the BS. Eq. 4.9 satisfies the constraint that only $M$ couriers are available.

**Table 4.1. ILP formulation for DIRSN.**

\[ \text{Minimize} \quad D_{\text{max}} \quad (4.3) \]

Subject to:

\[ \sum_{j \in N(i)} f_{ij} \leq t_i, \quad 1 \leq i \leq SN_{\text{total}} \quad (4.4) \]

\[ \sum_{j \in N(i)} f_{ij} + \sum_{j \in M(i)} l_{ij} \leq T_i, \quad 1 \leq i \leq C_{\text{total}} \quad (4.5) \]

\[ \sum_{j \in N(i)} f_{ij} - \sum_{j \in N(i)} f_{ji} = S_{G_i}, \quad 1 \leq i \leq SN_{\text{total}} \quad (4.6) \]

\[ \sum_{i \in M(j)} l_{ij} \leq c_j \sum_{1 \leq i \leq SN_{\text{total}}} S_{G_i}, \quad 1 \leq j \leq C_{\text{total}} \quad (4.7) \]

\[ \sum_{j \in N(i)} D_{\text{single}} \cdot f_{ij} + \sum_{j \in M(k)} D_{\text{single}} \cdot l_{kj} \leq D_{\text{max}}, \quad 1 \leq i \leq SN_{\text{total}}, 1 \leq k \leq C_{\text{total}} \quad (4.8) \]

\[ \sum_{i=1}^{c_{\text{total}}} c_i = M \quad (4.9) \]
We define Algorithm 1 to find candidate couriers that guarantee minimum delay to BSs among CNs and SNs and to execute the ILP. A link is only chosen if it offers minimum delay and satisfies the capacity constraint that is dependent on the packet generation rate per LN. End-to-end paths are determined based on the updates of routing tables between SNs and CNs. We note that Algorithm is applied only subject to the following two conditions: The availability of new data to be forwarded and the availability of suitable neighboring CNs. A suitable courier is one that passes by the SN at a speed that does not exceed a threshold $V_x$, which is equivalent to pedestrians and low-speed mobile objects. We also note that until the results of the ILP are returned, the algorithm will use existing routes. While this may result in utilizing suboptimal routes, our simulation results (as discussed in the next section) show that the algorithm is still superior to existing solutions. It should be also noted that the ILP in the DIRSN algorithm is executed in SNs and CNs, which are supposedly more computationally capable nodes.

Algorithm 1 (DIRSN): Finding candidates for minimum delay to access points among couriers and super nodes.

**Procedure FindMinDelayCandidates()**

set $Delay_{max} = \infty$

do for each triggered round

    for each super node $SN_i$ and courier node $CN_i$
do

        find all neighbours

        if $v_{neighbour} \leq V_x$

            exchange routing tables

        for each access point $BS_i$

            find set of neighbours with minimum $Delay_{API} < Delay_{max}$

            set $Delay_{max} = \min(\text{Delay}_{API})$

        end

    end

if (new data to forward && $v_{neighbour} \leq V_x$)

    call DIRSN ILP

end
4.2.3 Experimental Results

In this section, we evaluate the performance of DIRSN in practical settings, where the aforementioned probabilistic communication model is considered. We compare the DIRSN approach against the three RSN integration architectures discussed earlier (in Section 2.3), namely: TS, RS and MIX in terms of delay and delivery rate. We note that the placement ILP previously introduced for SIWR (in Chapter 3) is applied as well in the DIRSN simulations to assure optimal SN placement for our RSN-based topology. To compare the performance of the RSN architectures, the following three performance metrics are used:

1) Average delay, measured in msec and is defined as the amount of time required to deliver a data unit to the BS.

2) Average Packet Loss (APL), is the percentage of transmitted data packets that fail to reach the BS reflecting the effects of bad communication channels on data delivery across the architecture.

3) Average Generation Rate (AGR), is measured in Kbps. It is represented by the traffic rate generated by each SN or any equivalent node in a given architecture. In fact, AGR can be treated as a representation of the traffic generated by LNs in the network since it is a reflection of LNs count. In case of TS, RS, and MIX architectures, AGR at SNs will be the AGR of readers and sensors relaying to the couriers.

While studying these performance metrics, we vary the CN count to reflect the scalability of the exploited architecture under various applications.

4.2.3.1 Simulation Model

The proposed ILP-based approach is applied to the different integrated RSN architectures using MATLAB with settings similar to the previously described SIWR simulation (see Section 3.4).
The graphs’ dimensions are 500×500 m with 10 SNs and 100 LNs per SN on average. All mobile nodes representing CNs are set to follow the Random Waypoint mobility model [104]. We set the velocity threshold $v_x$ value to be 5 km/h which allows CNs represented by pedestrians, for example. The MATLAB simulator determines whether a wireless node is connected to its neighbors or not based on the previously mentioned probabilistic communication model, where the connectivity threshold $\tau = 70\%$. Each simulation experiment is repeated 500 times and the average results hold a confidence interval no more than 5\% of the average (of 50 runs) at a 95\% confidence (i.e. reliability) level.

4.2.3.2 Simulation Results

We apply the DIRSN ILP-based solution while increasing the CN count from 10 to 70 for each run. This range of CNs is applied to DIRSN and the other three integration architectures simultaneously and the resulting average delay in each of the four architectures is plotted against the corresponding count.

Figure 4.4 shows that for all integrated architectures, average delay tends to drop rapidly as the couriers’ count increases (from 10 to 70). This is only natural since an abundance of couriers will surely increase the connectivity over any network. We note, however, that even with as few as 10 CNs in the layout, our DIRSN approach scores an average delay that is 30\% better than the best average delay achieved by the other architectures. As the number of CNs increase to 70, DIRSNs’ average delay drops to 50\% of that achieved with 10 CNs. We attribute this to DIRSN’s distinct delay-tolerant courier selection algorithm.
Figure 4.4. Average Delay as CN count increases from 10 to 70 with 10 SNs and 100 LN/SN.

Figure 4.5 compares the average delay against APL percentage. Again, DIRSN produces the best (minimal) packet loss as delay increases (almost 85% of the generated packets are delivered when the average delay reaches 100 \( msec \)). DIRSN is more successful in providing alternative paths to deliver messages to their final destinations than the rest of the integrated architectures. This reflects the optimality of our placement and selection ILPs. Average delay and APL are always directly proportional. Yet, DIRSN provides a steadier and linear rate of delay-increase that remains substantially lower than the delay increase of the other three architectures, which all show an exponential rise in average packet loss beyond the 60 \( msec \) delay mark.
Lastly, we compare the average delay to the Average Generation Rate (AGR), measured in Kbps, in Figure 4.6. AGR is the traffic rate generated by the LNs that are associated to SNs in the network. As the delay increases, AGR does as well. However, Figure 4.6 also shows that DIRSN produces an increase rate that is more linear than the other architectures that experience, again, an exponential increase in their average delays as their corresponding AGR increases.

Figure 4.5. Average packet loss against average delay (40 CNs, 10 SNs, 100 LNs/SN).
The results as plotted in Figures 4.4 - 4.6 show that our DIRSN courier selection approach, coupled with our SIWR placement solution, outperforms all the three common RSN integration architectures in terms of delay delivery rates. It is worth mentioning that from the displayed results, MIX architecture always comes as a second-best to DIRSN, with a considerable gap between the two. We explain this as a result of MIX being the closest integration architecture to DIRSN in terms of treating sensors and tags as separate entities. DIRSN, however, excels in adopting an integration approach that is cost-efficient in terms of optimally deploying SNs and locating CNs in a way that better serves delay and delivery requirements.

**Figure 4.6. Average delay compared average generation rate (40 CNs, 10 SNs, 100 LNs/SN).**
4.3 Connectivity-Based Delivery Scheme

In this section, we present our Ubiquitous Robust Integrated Approach (URIA) for data delivery in RSNs. We note that connectivity between various parts of the IoT network is a consequence of seamless integration between its heterogeneous components. In this regard, we use the term *ubiquitous* to refer to the network’s ability to assume connectivity anywhere and anytime. Being *robust*, on the other hand, means that more than one path per each pair of nodes must be guaranteed at each data exchange cycle to avoid partitioning.

Loss of connectivity is common in harsh-environment applications, such as forestry fire detection or industrial plants monitoring, where the probability of node or link failure increases causing partitions in the network.

In general, connectivity problems [63], [105], [106], [107] can be dealt with either by populating relay nodes (RNs) or by utilizing mobile nodes. For example, in [106], the lowest number of relays is added to a disconnected static WSN, so that the network remains connected. In [107], mobile nodes are used to address $k$-connectivity requirements by identifying the least count of relays that should be repositioned in order to re-establish a particular level of connectivity. However, connecting nodes over IoT is more challenging because of the aforementioned mobility and partitioning issues and due to the high cost of deploying relay nodes. In addition, transmission on the IoT-scale may cover ranges exceeding the communication ranges of traditional WSN relays. To address this complexity, the authors of [63] propose a two-phase Optimized Relay Placement (ORP) approach that finds a set of candidate locations for relays in a way that maximizes the WSN connectivity. The connectivity of a connected graph $B$ is measured by the second smallest eigenvalue $\lambda_2$ of the Laplacian matrix $L(B)$ [92], where $\lambda_2$ indicates the minimum number of nodes, links or both whose removal (failure) would disconnect the graph.
The Laplacian matrix is a two-dimensional matrix that has -1 in the element \((i,j)\), if there is a connection between nodes \(i\) and \(j\), and 0 otherwise. It has the degree of node \(i\) in the element \((i,i)\). Given \(L(B)\), the algebraic connectivity of \(B\) is the second smallest eigenvalue \(\lambda_2\). Figure 4.7 shows a topology where partitioning is caused by the failure of at least three links (marked by dashed lines) or one node (marked in solid black).

![Figure 4.7. Partitioning scenario according to second smallest eigenvalue \(\lambda_2\).](image)

The approach we propose here achieves the desired level of connectivity by utilizing active motion (i.e. personal devices and vehicles) via CNs rather than passive recipients or relays. This method has several advantages in terms of reducing the cost of large-scale network deployment since we are utilizing ubiquitous and pre-existing resources. Also, CNs are mostly on the move, so their utility is manifested in different areas that would already be of higher population of users carrying courier resources.
4.3.1 System Models
URIA adopts an integrated RSN network model which has the same four components defined for DIRSN: Light Nodes (LNs), Super Nodes (SNs), Courier Nodes (CNs) and Base Stations (BSs) (see Section 4.2.1.1). The delay model is assumed here to be similar to the model proposed in Section 4.2.1.2.
As for the communication model, we assume a probabilistic model in which the probability of communication between two wireless devices decays exponentially with distance and takes into consideration surrounding obstacles and hindrances. This model can describe the path loss in the targeted site by taking into consideration the effects of the surrounding terrain on the power \( P_r \) of received signals as follows:

\[
P_r = K_0 - 10 \gamma \log(d) - \mu d
\]  

(4.10)
which follows a log-normal distribution centered around the average power value at the device location. Here \( K_0 \) is a constant incurred at transmission (of transceiver electronics), which is derived from the mean heights of the transmitter and receiver. Having \( d \) as the Euclidean distance between the transmitter and receiver, and \( \gamma \) as the path loss exponent, we adopt \( \mu \) as a normally distributed random variable with zero mean and variance, i.e. \( \mu \sim N(0, \sigma^2) \). Since the received signal could be quantified using \( P_r \), we devise a lower threshold on the signal level to deem communication successful. Denoting it as \( P_{min} \) over distance \( d \) (between transmitter and receiver), we denote the probability of successful communication as:

\[
P_c = P(P_r(d) \geq P_{min})
\]  

(4.11)

4.3.2 Problem Definition and Formulation
According to the aforementioned models, we define our deployment problem as follows:
Given a pool of candidate mobile/static N couriers in an IoT-based RSN architecture, select the best subset of N to deliver data from \( C_c \) SNs to a single BS while satisfying connectivity and delay constraints.

To solve this problem, we develop an SDP-based solution that determines the optimum subset of CNs in a manner that insures specific connectivity level with minimal delay across the network. We represent our layout by an initially connected graph, denoted by \( B \), whose vertices are the SNs and CNs in the setting. The algebraic connectivity of \( B \) is measured by the second smallest eigenvalue \( \lambda_2 \) of the Laplacian matrix \( L(B) \). Our solution aims toward satisfying a specific algebraic connectivity while minimizing the data delivery delay. We apply this method to other RSN deployment approaches and compare the resulting delivery rates to prove its superiority.

We assume discretized rounds from which our scheme selects routing paths. Selection is based on locating the most suitable couriers passing through these paths. This is determined according to minimum delay and connectivity-level constraints calculated from the SNs up to the BS. The location of the best \( M \) couriers in the proximity of each SN and available CN in the network is achieved by periodically exchanging routing tables and/or registration records between neighboring nodes. This location process is repeated at the beginning of each round. The mobility history of the neighbors is examined against the communication range of the corresponding destination node(s). Based on the results, forwarding candidates are defined according to minimum delays.

The solution we use to conduct this approach is based on an SDP formulation which, in turn, requires defining the following constants and variables:
Constants:

\( C_c \) is the total candidate couriers.

\( D_{\text{max}} \) is the maximum delay to transfer a data unit from a SN to the BS.

\( S G_i \) is the data generation rate of a SN \( i \) (based on the underlying connected LNs per SN).

\( t_i \) is traffic capacity of SN \( i \) (i.e. maximum bandwidth available for node \( i \) per round).

\( T_i \) is traffic capacity of CN \( i \).

\( N \) is the maximum couriers’ count that can be used per round.

\( N(i) \) is a set of neighboring candidates such that \( j \in N(i) \) if node \( j \) is within the transmission range of node \( i \) (i.e. \( P_r(i,j) \geq \tau \)).

\( M(i) \) is a set of indices such that \( j \in M(i) \) if node \( j \) is within the transmission range of a courier \( i \) that can reach a BS.

\( Q \) is a quality factor that is predefined based on the network specifications and user requirements.

\( n \) is the total connected SNs and BSs.

\( I_{nxn} \) is the identity matrix of size \( n \) by \( n \).

Variables:

\( \alpha_i \) is a binary variable equals to 1 when a courier at position \( i \) (associated with an (x, y) coordinate) is chosen by a SN to relay its data to the BS and 0 otherwise.

\( f_{ij} \) is the flow from a super node \( i \) to courier \( j \) (i.e. the data units to be sent from \( i \) to \( j \)).

\( l_{ij} \) is the flow from a courier node \( i \) to courier \( j \).

\( L(\alpha) \) is the Laplacian matrix of the connected graph formed by \( n \) SNs/BSs.

\( S \) is a scalar variable representing the 2nd smallest eigenvalue in \( L(\alpha) \).
Minimizing the delivery delay implies minimizing the total path length towards the BS without overwhelming the integrated network. This is achieved by locating a courier set that maintains the shortest path from each SN to a BS while considering their varying node/link capacities and load balance. We aim for each SN to deliver data to its corresponding BS with the least delay. In addition, this set of couriers must guarantee a specific level of connectivity ($\geq Q$) in the network formed by SNs and BSs. The SDP formulation in Table 4.2 represents our solution to address these constraints.

**Table 4.2. SDP formulation for URIA.**

Minimize $D_{\text{max}}$ 

Subject to:

\begin{align*}
    \sum_{j \in N(i)} f_{ij} & \leq t_i, \quad 1 \leq i \leq SN_{\text{total}} \\
    \sum_{j \in N(i)} f_{ij} + \sum_{j \in M(i)} l_{ij} & \leq T_i, \quad 1 \leq i \leq C_c \\
    \sum_{i \in M(j)} l_{ij} & \leq \alpha_j \sum_{1 \leq i \leq SN_{\text{total}}} SG_i, \quad 1 \leq j \leq C_c \\
    \sum_{j \in N(i)} D_{\text{single}} \cdot f_{ij} + \sum_{j \in M(k)} D_{\text{single}} \cdot l_{kj} & \leq D_{\text{max}}, \quad 1 \leq i \leq IN_{\text{total}}, 1 \leq k \leq C_c \\
    \sum_{i=1}^{c_c} \alpha_i & = N, \quad 0 \leq \alpha_i \leq 1 \\
    S \left( l_{nxn} - \frac{1}{n^{11}} \right) & \leq L(\alpha) \\
    S & \geq Q
\end{align*}
Eq. 4.12 is the objective function which minimizes $D_{\text{max}}$. Eqs. 4.13 and 4.14 satisfy the traffic capacity constraints available to SNs and CNs, respectively. Eq. 4.15 guarantees that if no courier is selected (i.e. $a_{ij} < 0.5$), no flow is sent to courier at position $j$. Eq. 4.16 makes $D_{\text{max}}$ the maximum delay over all SNs seeking the BS (note that we minimize $D_{\text{max}}$). Eq. 4.17 satisfies the constraint that only $N$ couriers are available. Eq. 4.18 formulates the mathematical representation of the minimum required nodes/links for a network to partition. Eq. 4.19 represents the connectivity-level constraint of the formulated network.

In this approach, we assume a connected graph constructed by the SNs and BSs. Connectivity of this graph is measured by considering its Laplacian matrix $L(\alpha)$ [63], [92]. The Laplacian matrix is a two dimensional matrix that has -1 at the element $(i,j)$, if there is a connection between nodes $i$ and $j$. It has an integer positive number at the element $(i,i)$ that represent the number of edges connected with node $i$. Given $L(\alpha)$, the graph connectivity (or algebraic connectivity) is mathematically measured by computing the second smallest eigenvalue $\lambda_2$ of the Laplacian matrix $L(\alpha)$, where $\lambda_2$ indicates the ratio of the minimum number of nodes/links whose removal would disconnect the graph to the total number of nodes/links in the graph.

By maintaining a $\lambda_2$ value that is greater more than a value $Q$, we assure a specific connectivity level in the formulated network (graph). This is due to the proportional relation between the value of $\lambda_2$ and the number of nodes/links which can cause network partitions. In order to maintain the $\lambda_2$ value, we assume $C_c$ candidates among the available CNs. We want to choose the optimum $N$ CNs amongst these $C_c$ candidates with respect to connectivity; where $N$ is constrained by a cost budget and/or available resources. We can then formulate this robustness objective using the two constraints in Eqs. (4.18) and (4.19). Then, we formulate and solve an optimized SDP with an objective function of minimizing the delay while assuring a specific connectivity-level ($\approx Q$).
without exceeding the available count of couriers from which we achieve our ubiquitous connectivity.

4.3.3 Experimental Results
Using MATLAB, we simulate randomly generated RSNs which have the graph topology proposed in the previous section and subject to varying Probabilities of Failure (PoF) reflecting the level of harshness in the deployment environment. To solve the previously modeled SDP optimization problem, we used the SDPA-M MATLAB Package [108].

4.3.3.1 Performance Metrics & Parameters
To evaluate our URIA approach against its rival, we use the following performance metrics:

1) Connectivity ($\lambda_2$): This criterion reflects the established network robustness under varying PoF. It gives an indication for the designed RSN efficiency in IoT settings.

2) Average delay: Defined as the time required to deliver a data unit to a BS.

3) Average Packet Loss (APL) percentage: The percentage of transmitted data packets that fail to reach the BS.

Three main parameters are used in the performance evaluation: Probability of Failure (PoF), Number of available couriers ($N$), and Average Generation Rate (AGR). PoF is the probability of connectivity failure in the network, due to node movement, node damage, link failure, etc. We chose this parameter as it is a key factor in reflecting the IoT settings in terms of heterogeneity and dynamics. As for $N$, it represents the availability of resources in providing improved connectivity alternatives. Lastly, AGR reflects the scalability and applicability of the proposed approach in large-scale and excessive data exchange applications.
4.3.3.2 Baseline Approaches

In order to evaluate URIA’s performance in terms of connectivity and delay, we run simulations that compare it against two delivery approaches. The first is ORP [63], an algorithm based on an SDP that maximizes the formulated network connectivity by maximizing $\lambda_2$. ORP opts to establish a robust network by selecting the most appropriate among a set of stationary relays. The second approach we compare against is DIRSN which we introduced in Section 4.2. The purpose of comparing URIA with DIRSN here is to demonstrate the variation in delivery of the two schemes given the differences between their approaches since DIRSN runs a delay-based solution as opposed to the connectivity-based solution of URIA which targets harsher settings with higher PoF in terms of nodes and links.

4.3.3.3 Simulation Model

The three deployment schemes: ORP, DIRSN, and URIA, are executed on randomly generated RSN graph topologies in order to get statistically stable results. The graphs’ dimensions are $500 \times 500$ m and contain 500 nodes, 300 of which are CNs, 20 are SNs and 10 LNs per SN. All CNs follow the Random Waypoint mobility model [104]. For each topology, we apply a random PoF values, and performance metrics are computed accordingly. A linear Congruential random number generator is used for random RSN. We assume a predefined fixed time schedule for traffic generation at the deployed SNs. Intermediate CNs are selected by applying the three approaches.

4.3.3.4 Simulation Results

We ran simulations to compare the three aforementioned approaches in terms of delay vs. connectivity, PoF vs. APL and AGR vs. APL. The average results hold confidence intervals of no more than 2% of the average values at a 95% confidence level. URIA is intended to achieve the
best of both ORP, which targets connectivity as an objective, and DIRSN, which specifies minimum delay as its objective function. It is apparent from Figure 4.8 that the three approaches demonstrate comparable outcomes for smaller values of $\lambda_2 (>0.02)$. However, as the connectivity constraint increases (i.e. harshness of the environment increases), both ORP and DIRSN lose the lead and maintain an almost steady delay, with DIRSN attaining a better score than ORP (48 msec vs. 68 msec). URIA, on the other hand, outperforms both approaches with a minimal delay that drops exponentially as connectivity increases.

![Figure 4.8. Delay vs. Connectivity for 300 CNs, 20 SNs and 10 LNs/SN.](image-url)
In Figure 4.9, we notice that DIRSN provides the best performance in terms of delivery for PoF < 20%. However, since DIRSN targets delay as its objective function, its feasible search space is limited by the connectivity constraint that increases as PoF rises. When PoF exceeds the 20% limit, DIRSN’s performance deteriorates in favor of URIA that reacts to both delay and connectivity, and eventually surpasses DIRSN. The same could be said about ORP that slightly outperforms URIA until the 10% PoF level. Beyond that point, the effect of URIA’s dynamic and delay-tolerant formulation gives it an advantage over ORP’s static-based algorithm.

Figure 4.9. Packet loss vs. Probability of Failure for 300 CNs, 20 SNs and 10 LNs/SN.
In Figure 4.10, URIA outperforms both ORP and DIRSN in terms of packet loss as packet generation rate increases. This reflects URIA’s scalability and applicability in large-scale and excessive data exchange applications. Again, we observe that ORP is a connectivity-based approach and it handles delivery poorly as AGR increases until it reaches a level where it drops 40% of its packets. DIRSN, being a delay-tolerant approach that incorporates a courier-based delivery algorithm, performs better than ORP under higher rates of data exchange.

Figure 4.10. Packet loss vs. Average Generation Rate for 300 CNs, 20 SNs and 10 LNs/SN.
4.4 Summary

In this chapter, we address delivery in integrated RSNs from two perspectives: first, we introduce in Section 4.2 DIRSN; an optimized Delay-tolerant approach for Integrated RFID-Sensor Networks. This is a delay-based novel scheme for data delivery and CN selection in RSNs. DIRSN considers the variations between nodes in IoT in terms of mobility and connectivity capacities. By associating these variations to the fact that an IoT setting is frequently disrupted, we employ CNs into a new decentralized ILP-based delay-tolerant approach that locates the optimum set of CNs per time-round. This approach aims toward guaranteeing minimum-delay-connectivity between Super Nodes (SNs) and Base Stations (BSs) by locating the best set of CNs capable of doing so without violating the main requirements of RFIDs and WSNs that are imposed by the dense deployment and node/link capacities such as load balancing. When compared against other RSN integration approaches, DIRSN performed superbly in terms of cost-efficiency, delivery rate and delay.

Second, we introduce in Section 4.3 URIA, a Ubiquitous Robust Integrated Approach for data delivery in integrated RSNs. URIA utilizes mobile couriers to maintain connectivity levels between integrated relay-reader nodes and their access points. This objective is common to deployments in harsh operational environments where failure rates are high. An SDP-based solution conceptualized on Laplacian matrix and $\lambda_2$ values was introduced to tackle the guaranteed connectivity objective. The SDP-based solution presented by URIA, on the other hand, minimizes the worst delay experienced between any SN/BS pair. Our simulations compare URIA against two integrated approaches, DIRSN and ORP which tackle delay and connectivity constraints in vast deployments, respectively. Simulation results show that our URIA excels in handling both objectives in terms of total latency and delivery rates.
Chapter 5

Priced Data Delivery

The data delivery schemes we introduced in Chapter 4 both employ ILP-based solutions to optimize courier selection in a topology that assumes offline settings before the network is operational. In such offline approaches, relaying plans are preset before the system is operational. Any updates on the presumed paths between the nodes require the reprogramming of the system. However, applications for large coverage areas with highly mobile and versatile nodes that are typical to IoT settings require dynamic online solutions that are more responsive to topology changes.

On a related note, Courier Nodes (CNs) in the previously discussed RSNs were assumed to participate in data delivery for no specific gain in return. We remark that part of the IoT vision dictates broadening the set of services provided to potential clients and the economic growth in the system as a whole [109]. Hence, data delivery, particularly from an IoT perspective, is of a participatory nature. To stimulate users’ willingness to contribute, incentives are often introduced to motivate participants. Numerous incentive-based schemes are proposed in the literature [110], [111], [112]. Yet, they are mostly intended for Peer-to-peer (P2P) and vehicular networks and none address the delivery metrics parameters for IoT-based topologies nor cater for heterogeneity in transmission standards and the consequent differences in data in terms of generation rate and priority.

In general, efficient pricing schemes for wireless networks capitalize on the differential values of each of the constituents. Following the basic laws of supply and demand, the abundance of resources and their homogeneity decrease their value. However, higher prices are usually
assigned to nodes with scarce services, or those with elevated Quality of Service (QoS) measures. Generally, factors such as bandwidth, buffering capacity, residual energy and tendency to maintain the social welfare of the system, all contribute to pricing.

We present in this chapter two priced data delivery schemes. The first is a Monetary Courier-based Relaying (MCR) scheme. This is a bottom-up scheme (i.e. from data sources to base stations) which defines policies to monetary compensate CNs (by SNs) for sharing their resources, such as buffer space and transmission energy, in the delivery process. However, an SN may choose to directly transmit its data to the BS at a higher cost (in terms of energy) if the data is particularly urgent or critical, hence, we propose a criticalness function for MCR to specify prioritized or delay-sensitive data.

The second scheme is top-down (i.e. from base stations to data sources). Here, we investigate the application of public sensing (PS) in an IoT-scale and propose a Priced Public Sensing (PPS) scheme where clients issue purchasing requests for specific data readings. Clients’ requests are guided by access points through a data cloud, and data prices are determined based on two factors: The data sources availability and a utility function that caters for the client’s demands (e.g. quality and delay). Figure 5.1 demonstrates the flow of data and payments in the two aforementioned priced delivery schemes.

The remainder of this chapter is organized as follows. Section 5.1 outlines the proposed approaches and highlights their contributions. Section 5.2 describes our MCR approach in terms of its system models, methodology and experimental results. Section 5.3 describes our PPS scheme in terms of its system models, methodology and experimental results. Section 5.4 concludes this chapter.
Outlines and Contributions

This chapter introduces two schemes for priced data delivery. Data pricing is determined by two entities: The data providing side which has its resource-based constraints for price specification, and the client who imposes quality constraints. As a remedy we propose a utility function to base the purchasing decision upon.

In terms of MCR, our contributions can be listed as follows:

![Figure 5.1. (a) Bottom-up priced delivery and (b) Top-down priced sensing.](image)
1) We introduce online heuristics detailing the mechanism of our (MCR) scheme in an integrated RSN architecture. Our scheme aims at enhancing delivery while conserving the system’s energy by relieving RRs from a portion of the relaying load.

2) We present a criticalness function within MCR that employs both RFID and sensory data attributes to prioritize forwarding of packets within an RSN topology.

3) We present a dynamic assignment of relaying price via intermediate couriers. As such, MCR allows each courier to arbitrarily decide its current “charge” for forwarding a data packet to a specific destination. Our pricing model caters for heterogeneity stemming from transmission limitations of couriers.

In terms of PPS, our contributions can be listed as follows:

1) We propose a framework for Priced Public Sensing (PPS) based on IoT-driven architectural model that integrates heterogeneous data sources in environments lacking an infrastructure.

2) We provide a dynamic multi-tier pricing scheme that, from the suppliers’ end, adheres to the social welfare of the PS system by incorporating lifetime and energy constraints while considering, from the consumers’ end, delay and quality metrics of the received data to insure maximum utility gain.

3) We provide heuristics for distributed data delivery that exploits the components of the aforementioned architectural model, including stationary and mobile data sources, in addition to our pricing schemes.
5.2 Delivery-Based Approach

We introduce in this section a set of heuristics defining an online implementation of a Monetary-based Courier Relaying (MCR) scheme for price negotiation and data forwarding in RSN architectures. Our heuristics are built upon the following core parameters:

1) Price as a parameter composed of two components. The first is courier price ($P_{CN}$) with respect to each CN. The second component is price threshold ($\tau_p$) per data packet; reflecting the maximum the data source may afford to deliver its data either via CNs or through direct transmission to the BS if possible.

2) Data Criticalness ($C_{data}$): A binary parameter assigned to each packet based on two RSN attributes: identity and sensed readings. Data with higher $C_{data}$ are better qualified for being forwarded to more costly CNs, or even directly transmitted to the BS.

3) Linger time ($T_{linger}$) is the residence time period for each courier entering the transmission range of a source node. $T_{linger}$ is calculated according to the courier’s reported trajectory. This, combined with $\tau_p$, will dictate if a CN is considered as a relaying node.

4) Energy ($E$). This parameter is effective on two levels. First, it affects $\tau_p$. If a packet is to be forwarded to a CN, minimum energy is calculated based on $T_{linger}$ in order to minimize energy lost on transmission. On the CNs’ level, on the other hand, the available CN’s energy is proportional to its $P_{CN}$.

5.2.1 System Models

In this section we describe the network, pricing and criticalness models that are the basis of our MCR heuristics. We then present our problem statement and the assumptions related to it.
5.2.1.1 Network Model

We employ in MCR the same integrated RSN architecture previously described for DIRSN (See Section 4.2.1.1). Figure 5.2 depicts the components of the integrated RSN network model. Note that the transmission lines are marked with data criticalness tuples which will be elaborated upon in section 5.2.2.

![Network model for MCR.](image)

5.2.1.2 Pricing Model

Our courier–based delivery price model has two components. The first is courier price \( P_{CN} \) with respect to each courier node. The second component is price threshold \( \tau_p \) per data packet; reflecting how much the source is willing to pay for its delivery either via couriers or through direct transmission. In fact, cost threshold and the aforementioned criticalness of data are used
here to differentiate a class of data which is not delay-tolerant and do not accept extended buffering and transmission durations. We refer to this class of data as delay-sensitive data.

**Definition 2. Delay-Sensitive Data (DSD)** is data whose validity, quality or both significantly decline as its delivery time $T_d$ increases. Hence, DSD is prioritized in terms of transmission order and cost.

The formulation of the two price components $P_{CN}$ and $\tau_p$ depends on factors such as couriers’ density, load and guarantee of delivery. For a given $CN_i$, $P_{CN_i}$ is a tradeoff of relaying for a monetary value based on the availability of its own resources to be provided in response to a relaying request issued by some $SN_j$. We specify four main parameters for each $CN_i$ to announce its $P_{CN_i}$ in reply to an SN’s relaying request:

1) Delivery time $T_d$ is a combination of the time the CN is on transit $T_{travel}$ and its pause time $T_p$, such that

$$T_d = T_{travel} + T_p,$$  \hspace{1cm} (5.1)

where $T_p$ is an averaged value calculated for each CN, reflecting the frequency and length of pauses along its announced path

$$T_p = \frac{\text{# of stops/trip}}{\text{avg duration/stop}}$$  \hspace{1cm} (5.2)

Hence, we define a normalized $T_d$ as

$$T_d' = \frac{T_d}{\max T_d}$$  \hspace{1cm} (5.3)

2) Buffer capacity ($C$). The CN’s capability of buffering (while performing SCF for DTD) is inversely related to its service price. We define a normalized buffer capacity for the set of CNs as
3) Transmission capacity \( T_X \) represents the transmission range for each CN. We normalize this parameter as

\[
T_X' = \frac{T_X}{\text{max}_T T_X}
\]

4) Energy \( (E) \). We adopt the general energy consumption model proposed in [67], in which energy consumed for receiving a packet of size \( S \) is

\[
E_{RX} = S \gamma,
\]

whereas the energy consumed for transmitting a packet of size \( S \) for distance \( d \) is

\[
E_{TX} = S(\epsilon_1 + \epsilon_2 d^\delta),
\]

where \( \gamma, \epsilon_1 \) and \( \epsilon_2 \) are hardware specific parameters of the utilized transceivers, and \( \delta \) is the exponent for path loss, which is the difference between the transmitted and received signal power.

Based on Eqs. (5.6) and (5.7), and the initial energy \( E_i \) of each node with its relative position to other nodes, we can calculate the remaining energy \( E_r \) per node after the completion of each operational round by

\[
E_r = E_i - T E_{TX} - R E_{RX} - A E_a,
\]

where \( T, R \) and \( A \) are the arrival rates of transmitted, received and aggregated packets per operational round, respectively, and \( E_a \) is the energy consumed for a single packet aggregation.

Based on these four main parameters we described above, we propose a cost function for each \( CN_i \) such that
As for the second courier-based cost component (i.e. \( \tau_{\text{cost}} \)), we note that its value is not determined by the CNs. Rather, it is calculated at each SN per data packet to be relayed such that

\[
\tau_p = Cost_{fc} + (Cost_{Tx} \times S)
\]

(5.10)

where \( Cost_{fc} \) is a flat charge, \( Cost_{Tx} \) is the transmission cost per byte and \( S \) is the packet size in bytes. Each packet will be assigned a cost-limit directly proportional to its criticalness.

### 5.2.1.3 Criticalness Model

Data criticalness (\( C_{\text{data}} \)), is a binary function determined by the SN based on the disjunction of sensed data and its ID. According to our RSN architecture, each SN will receive data from light nodes that are either tags, sensors, or combined ST nodes. The reader component of the SN will maintain a list of important tagged items (i.e. tag IDs) whose detection will mark a critical-tag instant (\( C_{\text{tag}} \)) (e.g. a VIP vehicle or a lost individual entering the premises). As for sensor nodes, criticalness of sensed data (\( C_s \)) is determined at sensor level by:

\[
C_{\text{sensor}} = \kappa \times \text{Normalized Sensed Data}
\]

(5.11)

where \( \kappa \) is a weighted factor that defines the criticalness of the sensed data based on average reading (e.g. extremely unusual temperature reading) or on spatial/temporal parameters (e.g. motion sensors triggered at a store after operation hours, or high Carbon Monoxide levels sensed within a residential area). In such instances, sensors send their readings to the relay component of SNs decide as critical. Accordingly, a SN will determine the criticalness of a data packet according to the conjunction function
\[ C_{\text{data}} = C_{\text{tag}} + C_{\text{sensor}} \]  

(5.12)

The criticalness of each data packet is hence determined with respect to a criticalness threshold \( \tau_c \) maintained by each SN to decide if the packet is worth paying \( \tau_p \), or if a less costly CN will do so. These scenarios are depicted in Figure 5.2 where immediate transmissions from SN1 to the BS are labeled with a high \( C_{\text{tag}} \) resulting in the tuple \((1, 0)\) whose disjunctions \((1 + 0) = \) critical, whereas transmissions with low criticalness are relayed via CNs \( a \) and \( b \), to the BS and SN2, respectively.

5.2.1.4 Problem Statements and Assumptions

Our online heuristics will adopt a data transfer scheme aiming towards solving the following problem:

*In an RSN topology of high density SNs and CNs, based on the criticalness of the data, each SN is to find the best choice of CNs to relay its data packets to prior to attempting to directly transmit to a BS.*

Based on the above statement, we assume that SNs are stationary and optimally deployed to cover all sensors and tags in its vicinity as discussed in Section 3. Each SN will periodically send a beacon announcing its location and the destination of its packet(s) in search for any CN within its interrogation zone. Accordingly, a CN will acknowledge the beacon only if it is willing to participate in data transfer. Any CN willing to participate in the delivery scheme has to know and announce its \( P_{\text{CN}} \), speed \( (V_{\text{CN}}) \), trajectory \( (J_{\text{CN}}) \) and final destination(s) \( (D_{\text{Dest}_{\text{CN}}}) \) whenever acknowledging an SN’s beacon. The final decision of utilizing any CN is left to the corresponding SN depending on the best it can get at a given point of time, not necessarily the
overall optimal values among all CNs. All the aforementioned assumptions are necessary in order to realize our MCR models and the corresponding transmission heuristics as follows next.

5.2.2 MCR Heuristics

We define the following parameters for our MCR algorithms that define the rules of price negotiation and packet exchange between SNs and CNs:

1) Handshake duration ($T_{ha}$): Time required to exchange acknowledgements between the SN and the CN.
2) Beacon interval ($T_b$): Time SN waits until retransmitting its beacon packet again.
3) Timeout ($T_{out}$): Total time SN waits until a given packet is carried by any CN before transmitting it directly to the AP with a price equivalent to $\tau_p$. This parameter depends on the criticalness of the packet.
4) Transmission time ($T_{tx}$) is set by the transceiver’s technology (e.g. WiFi, 4G, etc.) where

$$T_{tx} \propto \frac{\text{packet size}}{\text{data rate}}$$

(5.13)

We specify two algorithms at SNs and CNs to specify detail an SN will choose a specific CN as a courier based on the CN’s willingness to participate and its announced price, and how a CN utilizes its available resources to decide on the proper response to a SN’s request.

Algorithm 2 dictates the steps implemented by a given $SN_i$ that has a data packet to be transmitted with a threshold $\tau_p$. The value of $\tau_p$ is set in line 6 along with the packet’s $T_{out}$. If the data is critical then it is directly transmitted to the BS (lines 7, 8). Otherwise, the SN will broadcast a beacon to CNs that includes its location, message size and message destination and waits for reply (lines 12, 13). In case a reply is received, from a given $CN_j$, then the SN will
assign the packet to it only if $CN_j$’s price is less than the threshold (lines 14, 15). Otherwise, $SN_i$ will ignore $CN_j$ and wait for another courier node until the packet times out, in which case $SN_i$ attempts to directly transmit it to some BS (lines 16-19).

Algorithm 2: SN seeking CN to relay data load

1. Function $SN()$
2. Output:
3. $Accept_{msg}$: $CN_j$ has been selected for relaying
4. $Ignore$: $CN_j$ has been rejected
5. Begin
6. Initialize $P_{CN}, V_{CN}, I_{CN}, \tau_p$ and $T_{out}$ on current data packet
7. If $C_{data}$ then
8. Directly transmit to BS
9. $Ignore$ //reject courier
10. Else
11. Do
12. Broadcast beacon
13. Wait ($T_b$)
14. If Ack received AND $P_{CN} < \tau_p$ then
15. Return ($Accept_{msg}$) //assign packet to $CN_j$
16. While (not $T_{out}$)
17. If not assigned then
18. Directly transmit to BS
19. $Ignore$ //reject courier
20. End

Algorithm 3 dictates the steps taken by $CN_j$. Each CN will compute its $P_{tx}$ and $T_{linger}$ according to its speed and trajectory (line 8). If the CN receives a beacon (line 10) then it checks if the message’s destination matches one of it announced destinations or if it is passing by another SN that may help in relaying (lines 11-14). If so, then the CN waits for minimum distance between it and the SN in order to minimize energy loss on transfer according to Eq. 5.8. This entails waiting until the CN gets closer to the SN according to its calculated trajectory, or immediately relaying the data to the CN before it gets out of the SN’s transmission zone (lines 15-18). Either ways, the
values of $P_{tx}$ and $P_{CN}$ are sent to the SN which compares them against $\tau_p$. If an acknowledgement is received the data is transmitted to $CN_j$ and $SN_i$ is charged (Lines 21-23).

Algorithm 3: CN willing to relay packet from SN

1. Function $CN$ (msg)
2. Input:
3. $msg$: A message from $SN_i$ which is either a beacon or acceptance as a courier.
4. Outputs:
5. $P_{CN}, V_{CN}, J_{CN}, Dist_{CN}$
6. $Ack$: An acknowledgement in response to SN’s beacon
7. Begin
8. Initialize $T_j$ and $P_{in}$ of $CN_j$
9. Receive (msg)
10. If $msg = =$ beacon from $SN_i$ then // where $i \neq j$
11. Compute $T_{inger}$ in $SN_i$’s vicinity
12. If $(T_{inger} > (T_{hs} + T_{tx})$ AND $Dist_{SN_i} \in \{Dist_{CN_j}\} )$ OR $(SN_j \in \{Dist_{CN_j}\})$ then // CN meets other SN
13. If $J_{CN}$ satisfied then
14. Wait for min. $E$ position
15. Else
16. Transmit immediately
17. Set $P_{tx}$ and send to $SN_i$
18. Return $Ack$ with ($P_{CN}$)
19. If $msg$ received then
20. $Rx\_packet (SN_i)$
21. Charge $SN_i$
22. End

5.2.3 Experimental Results

Our packet-level simulations are run using mobile Ad-hoc networks of 1000 nodes (SNs and CNs) under a nominal bit rate of 2 Mbps. CNs move with a speed that is uniformly distributed between 0 and 120 km/hr according to the Random Waypoint model. In addition, mobility level is inversely proportional to pause periods that vary according to different traffic sources. Mobility, with varying velocities, and rest instances were restricted for all nodes within a rectangular 150x30 km grid. Experiments were run for pause periods of 100, 200, 300, 400, 500, and 600 sec.
in case of 1000 nodes. Communication was constrained to a range of 200 m. A CSMA technique with collision avoidance (CSMA/CA) was used to transmit packets [113]. The simulations use 100 and 500 traffic sources and a packet rate of 4 packets/sec. The $T_{oue}$ timer for MCR is set to 10 ms.

5.2.3.1 Performance Metrics

We evaluate two key performance metrics:

1) Packet delivery fraction; representing ratio of data packets delivered to the destination to those generated by the MCR sources, which reflects the degree of reliability of the routing protocol.

2) Average energy consumed, measured in milli-joules/byte. This metric includes all possible energy consumptions caused by transmission at the SNs, receiving at the CNs, retransmission at the CN, and receiving at the APs as defined in Eq. 5.8, where the constants $T$, $R$ and $A$ follow a Poisson distribution.

Energy may be viewed as a cost factor as well. We use this metric to show the quality of MCR in choosing better priced solutions.

5.2.3.2 Baseline Approach

We compare the data delivery scheme of our MCR scheme against two leading end-to-end data delivery algorithms for mobile Ad-hoc networks: Ad-hoc On-demand Distance Vector (AODV) [33] and New Reliable Routing Algorithm (NRRA) [114]. AODV is a reactive routing protocol that establishes a route on demand and maintains routes as long as they are needed by the sources. If a source node moves, end-to-end route discovery is reinitiated if it still requires a route to the corresponding destination. AODV uses a timer-based route expiry mechanism to promptly remove stable routes.
NRRA is a variant of Dynamic Source Routing (DSR) protocol [34] that uses source routing instead of relying on routing tables in intermediate devices. NRRA enhances the link failure in DSR by reducing the number of broken links. In this algorithm source chooses a stable path for nodes mobility by considering nodes position/velocity information. This algorithm can reduce the number of broken routes efficiently and can improve route stability and network performance effectively.

5.2.3.3 Simulation Results
Figures 5.3 – 5.6 represent the results of our simulation that compares our MCR scheme against AODV and NRRA in terms of packet delivery percentage and average energy consumed per measured in milli-joules/byte. Both metrics are plotted against pause time $T_p$ which is used here as a metric of mobility in the network. Figure 5.3 shows that MCR maintains the lowest energy consumption among the three schemes as the $T_p$ value increases for a topology of 100 SNs. This is also true as the number of SNs increases to 500 (Figure 5.4). This is expected since MCR heuristics utilize a pricing function that incorporates transmissions; either immediately to the APs or via CNs, based on criticalness/energy metrics. We note that the gap between AODV and MCR increases as the number of sources in the topology increase. This is due to AODV’s approach that maintains and repairs more end-to-end paths as more sources are added. This differs from MCR’s approach that utilizes available CNs regardless of the number of data sources.
Figure 5.3. Energy consumed vs. pause time for 100 SNs and 900 CNs.
Figure 5.4. Energy consumed vs. pause time for 500 SNs and 500 CNs.

Figure 5.5. represents the performance of the three schemes in terms of delivery fraction compared to $T_p$ for 100 SNs. Again, MCR performs better than the two rival delivery schemes. This distinction becomes considerably clearer as we increase the number of sources to 500 (Figure 5.6). The explanation of MCR’s superiority in this regard is embedded in its transmission function (Eq. 5.9) that better rewards couriers with lower $T_{id}$ (which incorporates $T_p$ according to Eq. 5.1). This maintains the delivery rate of MCR as the number of sources increase. Whereas the decrease in mobile CNs has a deteriorating effect on AODV, for instance, which experiences an 11.5% drop in its delivery rate (from 96% in Figure 5.5 to 85% in Figure 5.6).
Figure 5.5. Delivery fraction vs. pause time for 100 SNs and 900 CNs.
5.3 Sensing-Based Approach

The wide proliferation of wireless sensors has given rise to Public Sensing (PS) as a new data-sharing model. This vision can be extended under the umbrella of the IoT to include additional data sources like smart mobile devices and RFID tags. Henceforth, the term “sensor” is expanded in what follows to include any form of data-generating source that is either stationary or in transit, regardless of its communication standards or underlying technology.

Figure 5.6. Delivery fraction vs. pause time for 500 SNs and 500 CNs.
In this section we present a priced public sensing (PPS) scheme for heterogeneous IoT architectures. Our framework caters to service-based applications where data purchasing requests are initiated by human clients and routed, in a top-down manner, across a multi-layered architecture via a data cloud to multifarious data sources at the lower tier of the architecture. Accordingly, we introduce online heuristics for data delivery from both static and mobile sources. Since the delivery schemes introduced here are priced, we provide detailed pricing utility functions for data acquisition.

Our pricing functions observe the network’s resource limitations, supply/demand factors, as well as data constraints from the client’s perspective. We finally provide simulation results showing the efficiency of our data delivery scheme compared to other WSN and mobile ad-hoc schemes with respect to network size, lifetime, end-to-end delay and packet delivery ratio. Figure 5.7 depicts our view of such an IoT-driven PPS architecture, where data is generated by masses of Light Nodes (LNs) belonging to numerous peripheral networks, and delivered, possibly via some data collectors (DCs), to gateways (GWs) that interface each peripheral network with the data cloud and respond to the data requests addressed by Access Points (APs) on behalf of the clients. We note that the DCs in our proposed public sensing environment may as well comprise integrated RFID readers if the LNs within their trajectories include tags.
5.3.1 Systems models

In this section, we elaborate on our network, delay and lifetime models, in addition to explaining our pricing scheme and its utility function.

5.3.1.1 Network Model

We consider a multi-tier PS hierarchy with four main components: Access Points (APs), Gateways (GWs), Data Collectors (DCs) and Light Nodes (LNs) as depicted in Figure 5.8.

Figure 5.7. Architectural model of priced public sensing (PPS) in IoT.

96
At the top tier of our proposed architecture, APs receive data requests from clients and initiate data purchasing negotiations. Data resides at the lower tier of the architecture at Light Nodes (LNs) that include sensors, smart devices, RFID tags and other multifarious entities capable of providing PS data. Each group of LNs is assumed to form a sub-network of data sources according to their applications. Our scheme entirely dedicates LNs to data reading. Thus, data requests and consequent transmissions are not forthrightly conducted between APs and LNs. Instead, each LN delivers its sensed data by multi-hop transmissions through other LNs to one or more data collectors (DCs) that could be either stationary (SDC) or mobile (MDC), and is solely

Figure 5.8. Components of PPS Network model.
responsible for forwarding the LNs’ data load to a gateway (GW). GWs act as price intermediaries and interfaces between the peripheral network and the data cloud.

Since the deterministic mobility patterns of MDCs have an impact on both the network’s lifetime and delivery delay, we further elaborate on DCs by considering two special scenarios on a circular sensing field:

1) A network with a SDC located at the center of the sensing field. All data packets are destined to the data collector.

2) A network with a MDC that moves along the boundary (i.e. the perimeter) of the sensing field. Although we assume a circular boundary to simplify the analysis, it is important to note that our data delivery scheme is not limited to any shape regarding the sensing field or the trajectory of the MDC. We just consider this special case as a proof-of-concept and for the sake of quantitative analysis. Corresponding calculations for other shapes can be derived similarly.

For the MDC scenario, we define three types of LNs:

1) Regular LNs, which are not able to communicate directly with DCs.

2) Relay Nodes (RNs) that are closer to the field’s boundary (i.e. the MDC’s trajectory). Data packets are forwarded to RNs, where they wait for the MDC to collect them.

3) Fast Track Relay Nodes (FTRNs) that are marked by the MDC as the closest to the location of the GW. Such nodes are periodically identified by the MDC and are used for more delay-sensitive data as will be shown in Section 5.3.2.

Figure 5.9 shows an illustration of the two previously mentioned DC scenarios along with the aforementioned types of LNs.
In summary, we make the following assumptions regarding our network model:

1) All sensor nodes have the same data generation rate; every sensor node generates $M$ packets per time unit.
2) Sensor nodes are uniformly distributed over the sensing field.
3) The sensing field is a circle of radius $R$.
4) The data collector moves along the perimeter of the sensing field.
5) Sensor nodes consume $E_{tr}$ energy units to send one packet.
6) Every sensor node starts with an energy supply of $E_{init}$ energy units.

Figure 5.9. Illustration of two data collecting scenarios (a) SDC (b) MDC.
5.3.1.2 Lifetime Model

We define network lifetime as the amount of time until the data collector is unreachable i.e. there exists a LN that does not have a multi-hop path to the DC). Our comparison assumes that the load is evenly distributed over RNs. We also ignore the energy consumed to receive data since transmission is known to be the dominant energy consuming operation [25]. In the following, we compare the two aforementioned data collecting scenarios in terms of network lifetime. We note that this comparison is based on the work presented in [115].

Stationary Data Collectors

With multi-hop communication, the lifetime of the network is determined by the lifetime of RNs (i.e. nodes whose distance to the SDC is not more than $r$ m). Since sensor nodes are uniformly distributed over the sensing field, the number of relaying nodes is expressed as

$$n = \frac{n\pi r^2}{\pi R^2} = \frac{nr^2}{R^2}$$

(5.14)

where $n$ is the total number of sensors (LNs) and $r$ is the transmission range of all the LNs. These RNs are in charge of delivering all data generated within their designated area to the SDC. Therefore, they transmit $Mn$ packets to the data collector. With a perfect load balancing, each relaying node transmits

$$\frac{MnR^2}{nr^2} = \frac{MR^2}{r^2}$$

(5.15)

packets. Thus, the lifetime of a relaying node is

$$\left[ \frac{E_{init}}{MR^2E_{tr}} \right] r^2$$

(5.16)

time units.
Mobile Data Collectors

Since the data collector moves along the perimeter of the sensing field, we have

$$\frac{\pi (R^2 - (R - r)^2)n}{\pi R^2} = \frac{(2Rr - r^2)n}{R^2}$$ (5.17)

relaying nodes (RNs). With a perfect load balancing and a constant data generation rate, each relaying node will be in charge of transmitting

$$\frac{MnR^2}{(2Rr - r^2)n} = \frac{MR^2}{(2Rr - r^2)}$$ (5.18)

packets. Thus, the lifetime of a relaying node in time units is

$$\left[ \frac{E_{init} (2Rr - r^2)}{MR^2 E_{tr}} \right]$$ (5.19)

From this comparison, the lifetime of a network with a mobile data collector is longer than that of a network with a stationary data collector by a factor of

$$\frac{2R - r}{r}$$ (5.20)

For example, if $r=100$ m and $R=1000$ m, the lifetime of a network with a mobile data collector is 19 times longer than the lifetime of the same network with a stationary data collector. We also note that the amount of energy consumed to send a packet over a multi-hop path is also proportional to the number of hops and, hence, can be approximated to a linear function of the Euclidean distance between the source and the destination. Therefore, the maximum amount of energy consumed to send a packet to a stationary data collector is $\theta R$ energy units, where $\theta$ is a constant. On the other hand, the maximum amount of energy consumed to send a packet to a MDC is $\theta R^* \text{ energy units}$, where $R^*$ is the maximum distance between an LN and its nearest RN. $R^*$ is expected to be smaller than $R$. 

101
5.3.1.3 Delay Model

For a network with a SDC (Figure 5.9-a), the delay a packet encounters is proportional to the number of hops between the packet's source and the SDC. The number of hops between two points can be approximated to be a linear function of the Euclidean distance between them. Therefore, the maximum delay a packet encounters is linearly proportional to the diameter of the network. This results in a delay of $\theta R$ time units, where $\theta$ is a constant and $R$ is the maximum distance between the SDC and a LN.

On the other hand, the maximum delay in a network with a MDC (Figure 5.9-b) is significantly worse. With a MDC, the worst case occurs when a packet arrives to a RN which has just been left by the MDC and it is the first node facing the MDC when it exits the communication range of a GW; such a packet needs to wait for the MDC to complete two full rounds over its trajectory which depends on the speed of the MDC. Let $Max_D$ denote the maximum delay a packet may encounter waiting for a MDC at some RN in the network. Apparently $Max_D$ is expected to be greater than $\theta R$.

To summarize, although having an MDC can potentially prolong the lifetime of the network and save energy, it causes a larger delay as compared with an SDC. This brings together a group of competing objectives and makes a demand for a reciprocal optimization scheme to choose the right data gathering strategy, and this is the main motivation for this work.

5.3.1.4 Data Delivery Model

Delivery is based on required data’s delay constraints. We thus have two delay-based delivery models: Delay-tolerant and fast track (or delay-sensitive) delivery.
Delay-tolerant Delivery

In order for LNs to deliver their delay-tolerant data to the data collector, they need to have a path to at least one relaying node. To do so, RNs broadcast their identity at the deployment stage and each sensor node keeps a record of the next hop towards a relaying node.

Fast Track Delivery

Delay tolerant routing prolongs the lifetime of the network by balancing the load over the network by sending to multiple destinations (i.e. RNs according to our model) rather than to a single bottleneck LN. On the other hand, data packets encounter significant delay before reaching the gateway. This delay, which is dominated by the physical motion of the MDC, has two factors. The first is the speed of the MDC. The second is the distance travelled by the MDC between the RN, where the data is picked up, and the gateway. While controlling the speed of the MDC is out of the scope of our work, we can minimize the travelled distance by deliberately targeting the RNs that are very close to the gateway as Fast Track RNs (FTRNs) (see Figure 5.9-b). This could make a significant difference especially if we have multiple MDCs which is expected in public sensing.

5.3.1.5 Pricing Model

The PPS scheme we propose is priced. That is to say, data is delivered upon client’s request in exchange for a monetary charge or a cost (in terms of transmission resources) to be compensated for by the requesting party. The data source/provider, on the other hand, sets a price to its data based on attributes related to resource availability such as energy, transmission capacity and the lifetime of the corresponding peripheral network. This latter part of the pricing scheme is conducted by the GWs In addition, our scheme caters to the users’ quality concerns by adhering to a utility function that is adjustable to different data types and quality parameters.
We hence provide a two-tier dynamic pricing scheme. At the lower tier, GWs reply to AP data requests with a resource-based service price. At the top tier, the requesting AP generates a utility function based on the user’s price limit and the GWs replies to decide on the chosen (paid) data source.

**Gateway Pricing Scheme**

We propose a decentralized approach where each gateway (\(GW_i\)) decides on its data/service price (\(P_{GW_i}\)); a tradeoff of relaying for a monetary value based on the availability of its own resources.

We specify three main parameters for each \(GW_i\) to announce its \(P_{GW_i}\) in reply to an AP’s request: Delay, capacity and lifetime.

Delay (\(D_{GW}\)), is a combination of the time (\(T_e\)) a RN will need to transmit an available data packet and the time the sensed data will need to arrive to the GW (\(T_{GW}\)) such that

\[
D_{GW} = T_e + T_{GW} \tag{5.21}
\]

Hence, we define a normalized \(D_{GW}\) as

\[
D'_{GW_i} = \frac{D_{GW_i}}{max_D} \tag{5.22}
\]

where \(max_D\) is the maximum expected delay.

As for relaying capacity (\(C_{GW_i}\)), we note that our delivery scheme adopts a data delivery approach where GWs have a limited capacity for the maximum amount of data that can be relayed over a specific time period. The GW’s capacity is directly related to its service price (\(P_{GW}\)). We define a normalized relaying capacity for the set of GWs as

\[
C'_{GW} = \frac{C_{GW_i}}{max_C} \tag{5.23}
\]

where \(max_C\) is the maximum expected capacity.
As for lifetime \((L_{GW_i})\), we adopt the general energy consumption model proposed in [64] to evaluate the discussed network lifetime in Eq. 5.16.

The relation between the aforementioned parameters is expressed according to the following function

\[
P_{GW_i} = \Phi \frac{C'_{GW_i} L'_{GW_i}}{D'_{GW_i}}
\]

where \(\Phi\) is a weight variable.

**Client Utility Function**

In a service-based model, a client may initiate data requests from data sources through APs and GWs, respectively. In this top-down process, data access price is to be provided by the entity requesting the service. This price is not to be confused with the previously mentioned price of the delivery-based model in Section 5.2. Data delivery price may be viewed as a “service price tag” from the supplier’s viewpoint, while the data access cost is the “fair purchasing price” from the consumer’s viewpoint.

The client/consumer’s estimate on the cost of the data obeys a utility function designed to cater for specific quality parameters in addition to the type of data requested by the client. In order to construct our utility function, we categorize the data to be provided within our PPS scheme into four main types:

1) Hard Real-time bounded Multimedia (HRM)
2) Soft Real-time bounded Multimedia (SRM)
3) Delay-Tolerant Data (DTD)
4) Delay-Sensitive Data (DSD)
The first two types (HRM and SRM) are inspired by the work presented in [116], [117] based on the type of end-to-end Quality of Service (QoS) guarantees to be expected with voice and video data transmitted either in real-time (HRM) or buffered (SRM). DTD and DSD, on the other hand, are suggested here for non-multimedia data (e.g. alert signals, geographic coordinates, environmental measures, etc.) which allow more relaxed constraints on delay and quality constraints. Nevertheless, some non-multimedia could have tight delay requirements (e.g. medical alerts). Henceforth, we define the following three utility parameters for each data request depending on its type:

1) Delay Sensitivity \((DS)\) of the requested data. The AP may mark some requests to be delay-sensitive or delay-tolerant; and the user may be willing to pay more for less delay, or to compromise some delay for a better (lower) price.

2) Quality \((Q)\), representing the level of quality of transmission in an aggregated normalized term (e.g. bit-rate or average packet loss) [118]. Quality can also be defined with respect to reliability of the source (i.e. in terms of proximity to the sensed phenomenon) or with respect to the data’s worth to the user. Some services (i.e. VoIP and video streaming) may require a higher quality level in terms of transmission rate compared to traffic updates, for instance.

3) Trust factor \((T)\) is a history-based function that is calculated at the AP per GW to represent a \(GW_j\) fulfillment measure. A higher \(T_{GW_j}\) indicates that previous data exchanges between \(AP_i\) and \(GW_j\) have been fulfilled according to the requirements (e.g. delay) promised by \(GW_j\).
Consequently, when a data request \((Data)\) is established, the corresponding base station \((AP_i)\) sets a reservation price \((P_r)\) to that request consistent with the client’s service agreement plan, such that \(P_r\) defines the maximum price a client is willing to pay for \((Data)\), where:

\[
P_r = \frac{Data_{\text{value}}}{Data_{\text{scarcity}}}\tag{5.25}
\]

such that

\[
Data_{\text{scarcity}} = \frac{\text{no. of replying gateways} + \xi}{\text{total no. of gateways}}
\tag{5.26}
\]

where \(\xi\) is a small constant chosen so that \(0 < D_{\text{scarcity}} \leq 1\), in order to avoid a situation where \(P_r = \infty\).

If \(GW_j\) replays back, its reply to \(AP_i\) will include the following parameters:

1) Its service cost \((P_{gw_j})\) as specified in Eq. 5.24.

2) The level of expected Delay \((D_{gw\_j})\) its reply would take, as specified in Eq. 5.21.

   Accordingly, the \(AP\) will use this parameter in its delay utility component according the function

   \[
   1 - e^{\frac{-\alpha}{D_{gw\_j}}}\tag{5.27}
   \]

   where \(D'\) is the normalized delay.

3) The level of quality \((Q)\) it is willing to provide based on its resources. Accordingly, the BS will use this parameter in its quality utility component according to the function
\[ \frac{1}{1 + e^{-\epsilon(Q'_{GW_j} - \beta)}} \quad (5.28) \]

where \( Q' \) is the normalized quality.

In addition, upon receiving the reply, the AP will generate a Trust utility factor \((T)\) per SN. The calculation of \((T)\) could follow a function similar to the fuzzy reputation formula presented in [119]. Eqs. 5.24, 5.27, 5.28 and the value of \( T_{SN_j} \) are used by \( AP_i \) to determine the utility score of \( GW_j \) after they are tested against the two constraints: \( D_{GW_j} \leq DS \) and \( P_{GW_j} \leq P_r \).

Hence, we define our utility function based on the aforementioned formulation as:

\[
U_{GW_i} = \left[ \frac{1 - e^{-\alpha P_{GW_i}}}{P_{GW_i}} \right] \times (T_{GW_i})^\sigma
\quad (5.29)
\]

The function in Eq. 5.29 utilizes the aforementioned utility parameters after normalization, in addition to \( (P_{GW_i}) \) in a manner that maps the expected user experience to changes in individual utility parameters.

To show the impact of each utility parameter on our utility function (Eq. 5.29), we present the plots in Figures 5.10 – 5.12 for each of the utility parameters \( D, Q \) and \( T \), respectively. In Figure 5.10, \( D \) is plotted with a constant \( \alpha \) that determines the rate of decrease of the exponential utility in Eq. 5.27. This particular function was chosen for \( D \) to reflect the rate of loss in the Quality of Experience (QoE) [120] as delay increases. By varying the value of \( \alpha \), it is possible to achieve different levels of delay-tolerance as shown in Figure 5.10, where we chose \( \alpha = 0.5 \) for delay-tolerant data and \( \alpha = 0.1 \) for more delay-sensitive data. We note that, for a delay-sensitive data request, a very low delay has to be achieved in order to provide a high delay utility component.
The Quality parameter ($Q$) is plotted in Figure 5.11. We note that we adopt a Sigmoid function according to Eq. 5.28 where the tolerance to variation in data quality is expressed by fixing the value of $\epsilon$ (set here to 10) and varying the inflection point denoted by the value of $\beta$. Thus, if the requested data is quality-sensitive (e.g. VoIP is sensitive to low transmission rate) the function will require a higher value of $Q$ before the utility increases (as depicted in the lower plot with $\beta = 0.8$ in Figure 5.11). In contrast, lower constraints on quality require a utility that increases rapidly at a lower value of $Q$, which can be achieved with an early inflection point ($\beta = 0.5$ in

![Figure 5.10. Delay function plot.](image)
Figure 5.11. The value of Q in Figure 5.11 ranges from 0 to 1, where 1 indicates the best level of quality attainable depending on the quality metric.

Lastly, Figure 5.12 shows the plot of the Trust function where $T \in [0,1]$. Note that an AP can give more emphasis to this parameter through the factor $\sigma$ to particularly penalize gateways with bad service accounts. This is shown in the lower plot of Figure 5.12 where $\sigma = 2$, whereas the upper (better $T$) plot is a result of $\sigma = 1$. 

![Figure 5.11. Quality function plot.](image)
We note here that we multiply the Trust component by the rest of the utility function in Eq. 5.29 in order to balance the effect of “deceiving” gateways that may offer attractively low $P_{GW}$ in order to pass false quality promises. Moreover, we divide the utility function by $P_{GW}$ in order to protect the client from situations where two or more gateways happen to achieve almost equal utility scores while charging prices that, although less than $P_r$, largely vary.

We provide three 3-D graphs in (Figures 5.13 – 5.15) representing the interactions between the different combinations of individual utility components as expressed by Eqs. 5.27 - 5.29. For

Figure 5.12. Trust function plot.
example, Figure 5.14 shows how the combined Trust-Delay utility varies with the individual values of $D$ and $T$. To achieve a high combined utility score with this specific combination, $T$ has to be high (maximum $T = 1$) while $D$ has to remain low.

Figure 5.13. Delay vs. Quality 3-D rendering.
Figure 5.14. Trust vs. Delay 3-D rendering.
To demonstrate the impact of the utility parameters on the type of requested data in our PPS scheme, we provide in Figures 5.16 and 5.17 plots of utility functions defined by Eq. 5.29 for the four data types: HRM, SRM, DSD and DTD while varying delay and quality requirements, respectively. We set the following values per data type:

**HRM:** \( \alpha = 0.1, \beta = 0.85 \)

**SRM:** \( \alpha = 0.5, \beta = 0.6 \)
DSD: \( \alpha = 0.3, \beta = 0.35 \)

DTD: \( \alpha = 0.9, \beta = 0.2 \)

The values of \( \sigma \) and \( \epsilon \) are fixed for all data types at 2 and 10, respectively. The values mentioned above were chosen based on the plots in Figures 5.10 – 5.12 to reflect the nature of each data type, where HRM is the most sensitive in terms of quality and delay, whereas DTD is the most tolerant among the four types in terms of quality and delay.

Figure 5.16. Utility Vs. Delay for different data types.
In Figure 5.16, we note that the four plots follow the benchmark trend depicted in Figure 5.10. HRM is the strictest in terms of delay component. The utility plot of HRM deteriorates faster than the rest of data types at $D = 0.5$ msec. DTD is the most tolerant in terms of delay, achieving a higher utility score throughout the delay axis.

![Utility Vs. Quality for different data types.](image)

**Figure 5.17. Utility Vs. Quality for different data types.**

Figure 5.17 shows how the four data types react to quality requirements with respect to our utility function. Again, HRM and DTD are impacted the most by quality sensitivity. As for DSD, we see that it is more lenient than SRM, which is represented by buffered video streams, for instance.
DSD reaches a utility saturation level faster (at $Q = 0.75$) whereas SRM continues to demand utility to serve its quality requirement.

5.3.2 PPS Heuristics

According to the PPS system model, the targeted PPS scheme aims towards the following:

*Given an abundance of multi-owned APs, GWs, and MDCs; APs are to find the best GWs to cater for their data requests within specific geographical vicinities while considering different utility parameters for different types of sensed data.*

We address, hence, two problems:

1) Finding a mechanism to manage the relationship between APs and GWs at the top-tier of the network model. This mechanism is supposed to help each AP to pick the right GW to be used for each data request according to a particular set of utility parameters.

2) Finding the routing infrastructure for the lower-tier of the model that supports delivery of both delay-tolerant data and delay-sensitive data to the AP.

The following two subsections explain our methodology to tackle each of the aforementioned problems.

5.3.2.1 Top-Tier Gateway Selection

We specify two algorithms at APs and GWs. These algorithms detail how an AP will send a data request and how it will process the parameters received from acknowledging GWs into the utility function. The algorithm at the GWs level sets the parameters to be included in a reply to an AP’s request and the how each GW will determine its service (data) price according to the availability of its resources.
Algorithm 4 specifies the steps of a query issued by an AP seeking data (for a client’s service request) from any GW in the set \{GW_0, ..., GW_n\}. This set is determined according to the geographical vicinity (GV) of the request (line 9). As mentioned earlier, the AP waits for the GW acknowledgements and basis its selection decision on the returned utility parameters (lines 12, 13), in addition to \(P_{GW_i}\) and \(T_{GW_i}\) which is calculated by the AP (line 14). Then, the utility score of each GW is calculated according to type of data and required services (lines 16-18). The AP pays the price finally to the chosen GW with a request to release the data (lines 20, 21).

---

**Algorithm 4: AP seeking GW to provide priced data**

1. Function \(AP\) (Client_Req.x)
2. Input
3. Client_Req.x: A client’s data request for some service \(x\) to be provided by this AP.
4. Output:
5. \(GW_i\): A selected \(GW_i \in \{GW_0, ..., GW_n\}\) to pay and receive data from.
6. Begin
7. Set Data, Data.type //define requested data and its type according to four predefined types
8. Set \(DS, P_r\) for Data
9. Initialize GV, GW{ } //geographic vicinity and set of replying GWs
10. Request Data from GWs in GV
11. While an Ack is received from \(GW_i\)
12. If \(P_{GW_i} \leq P_r \) and \(D_{GW_i} \leq DS\) then
13. Add \(GW_i\) to GW{ }
14. Calculate \(T_{GW_i}\) // Equation (5.29)
15. End
16. Based on Data.type do //service utility preferences
17. For each \(GW_i \in \{GW_0, ..., GW_n\}\) do
18. Calculate \(U_{GW_i}\) //Equation (5.29)
19. End
20. Select \(GW_i\) with best \(U_{GW_i}\)
21. Pay \(P_{GW_i}\) to \(GW_i\)
22. End

---

118
Algorithm 5 shows how a GW responds to an AP data request. The GW validates the request on two levels. First it checks if the requested data is available within the data sources (LNs) under its immediate coverage (lines 8-11). If this was the case, then the delay component $T_{GW_i}$ in Eq. 5.21 is expected to be zero and the whole delay factor is hence drastically lower than the following scenarios (lines 12-14) where the GW validates the data request with the MDCs within the specified GV. In either case, whenever the data is found, GW-corresponding utility parameters are assigned to it according to Eqs. 5.24, 5.27, 5.28 and 5.29 and are included in an acknowledgement to the AP request (lines 11 and 14). Once the AP accepts the offer, the data is sold to it (lines 17-19).

Algorithm 5: GW reply to AP query request

1. Function $GW (Data)$
2. Input:
3. $Data$: A data request from $AP_i$.
4. Output:
5. $RA$: Request answer that could be an Acknowledgement to the Data request including Price, Quality and Delay parameters.
6. Begin
7. Initialize list of static and dynamic sensors in GV
8. If requested $Data$ is available at static sensors then
9. Set $Delay = 0$
10. Set $P_{GW}$ //Equation 11
11. Return ($RA=Ack$) with $P_{GW}$, available $Q_{GW}$
12. Else if requested data is available at mobile sensors then
13. Set $P_{GW}$ //according to different Delay expectation
14. Return ($RA=Ack$) with $P_{GW}$, available $Q_{GW}$ and expected $D_{GW}$
15. Else
16. Ignore
17. If this $GW$ is selected by $AP_i$ then
18. Return ($QA=Data$)
19. Charge $AP_i$ with $P_{GW}$
20. End
5.3.2.2 Lower-Tier Data Delivery

In order for LNs to deliver their delay-tolerant data to the data collector, they need a path to at least one RN. Data is stored in the relaying node until an MDC passes by to pick the data up. The data is then carried by the MDC until it becomes within the communication range of a gateway; at that point the MDC delivers the data to the gateway, which makes it available to users. To do so, RNs broadcast their identity at the deployment stage and each sensor node keeps a record of the next hop towards an RN. Each LN $n_i$ has a Relaying Node Record $RNR_i$ which has the following fields:

1) $id$: the id of the relaying node to which delay-tolerant data will be sent.

2) $Next\_hop$: a neighbor of $n_i$ which is used as a next hop towards the relaying node.

3) $Number\_of\_hops$: the number of hops to the relaying node.

For routing delay-tolerant and delay sensitive data, Algorithm 6 describes the process of setting the FTRNRs and RNRs of all sensor LNs, assuming that each LN uses the nearest relaying node. Note that this process will construct a tree for each relaying node; the tree of a relaying node $n_i$ is rooted at $n_i$ and involves all sensor nodes whose nearest relaying node is $n_i$. FTRNRs and RNRs will be identified at the initialization of the network.
Algorithm 6: Delay-tolerant and fast track routing records

Function $R\text{N}(\ )$
For each $L\text{N} \ n_i$ do
  If $n_i$ is a $F\text{T}\text{R}\text{N}$ then
    $F\text{T}\text{R}\text{N}i$.id=$i$;
    $F\text{T}\text{R}\text{N}i$.next hop=$i$;
    $F\text{T}\text{R}\text{N}i$.number of hops=0;
    broadcast $F\text{T}\text{R}\text{N}i$ to all neighbors of $n_i$ ;
  ElseIf $n_i$ is a $R\text{N}$ then
    $R\text{N}i$.id=$i$;
    $R\text{N}i$.next hop=$i$;
    $R\text{N}i$.number of hops=0;
    broadcast $R\text{N}i$ to all neighbors of $n_i$ ;
  Else
    $R\text{N}i$.number of hops = $N + 1$;
  End
End

When a $L\text{N} \ n_i$ receives a broadcasted $R\text{N}j$ :
  If $R\text{N}j$.number of hops + 1 < $R\text{N}i$.number of hops then
    $R\text{N}i$.number of hops = $R\text{N}j$.number of hops + 1;
    $R\text{N}i$.id = $R\text{N}j$ .id;
    $R\text{N}i$.next hop = $j$;
    broadcast $R\text{N}i$ to all neighbors of $n_i$ ;
  End
End
When a $L\text{N} \ n_i$ receives a broadcasted $F\text{T}\text{R}\text{N}j$ :
  If $F\text{T}\text{R}\text{N}j$.number of hops + 1 < $F\text{T}\text{R}\text{N}i$.number of hops then
    $F\text{T}\text{R}\text{N}i$.number of hops = $F\text{T}\text{R}\text{N}j$.number of hops + 1;
    $F\text{T}\text{R}\text{N}i$.id = $F\text{T}\text{R}\text{N}j$ .id;
    $F\text{T}\text{R}\text{N}i$.next hop = $j$;
    broadcast $F\text{T}\text{R}\text{N}i$ to all neighbors of $n_i$ ;
  End
  If $F\text{T}\text{R}\text{N}j$.number of hops + 1 < $R\text{N}i$.number of hops then
    $R\text{N}i$.number of hops = $F\text{T}\text{R}\text{N}j$.number of hops + 1;
    $R\text{N}i$.id = $F\text{T}\text{R}\text{N}j$ .id;
    $R\text{N}i$.next hop = $j$;
    broadcast $F\text{T}\text{R}\text{N}i$ to all neighbors of $n_i$ ;
  End
End
5.3.3 Experimental Results

The proposed sensor environments were simulated using MATLAB. We compare the data delivery scheme of our PPS framework against Ad-hoc On-demand Distance Vector (AODV) [33] and New Reliable Routing Algorithm (NRRA) [114]. We remark, however, that we employ here modified versions of AODV and NRRA. This is because the original algorithms specify end-to-end routes, while the versions we use here are oblivious on the link state within the data cloud. We define this as the cloud effect. In other words, they are restricted by their corresponding gateways and their view of the topology is limited by their own peripheral networks. Hence, we will refer to AODV and NRRA through the reminder of the section as PODV and PRRA, respectively, where the ‘P’ stands here for Peripheral.

We adopt the energy consumption model proposed in [25] which is described here as follows:

\[ E_{Tr}(r, B) = b \times (e_{elec} + e_{amp} \times r^Y) \]  \hspace{1cm} (5.30)

\[ E_{Rc}(b) = b \times e_{elec} \]  \hspace{1cm} (5.31)

where \( E_{Tr} (r, B) \) is the energy consumed to send \( b \) bits over \( r \) meters, \( E_{Rc} (b) \) is the energy consumed to receive \( b \) bits, \( e_{elec} \) is the energy consumed by the transmitter (receiver) to send (receive) one bit, \( e_{amp} \) is the energy consumed by the transmission amplifier for one bit, and \( Y \) is the path-loss exponent. In our simulation, \( r \) is set to 50 m, \( e_{elec} \) is set to 50 mJ/bit, \( e_{amp} \) is set to 0.1 mJ/bit/m^2, and \( Y \) is set to 2. The packet size is 512 bits. Every sensor node has an initial energy of 50 J and generates 150 pkts/round. A round is defined as the time span per which all LNs would have reported their targeted data. Our simulations involve networks of 2000, 4000, 6000, 8000, 10000, 12000 and 14000 sensors randomly deployed in a 300×300 m^2, 400×400 m^2, 500×500 m^2, 600×600 m^2, 700×700 m^2, 800×800 m^2 and 900×900 m^2 fields, respectively. We
also simulated environments for a fixed GW count of 100 and LN count of 4500 while varying the data DC count from 5 to 30 in increments of five. Each RN has, for simplicity, a fixed relaying capacity equal to 50% of its generated traffic. For each network size, we test 20 instances and take the average. To generate a trajectory for the data collector, we use a simple method. We divide the sensing field into four equal-size squares; the trajectory of the data collector is a quadrilateral that has a vertex inside each square.

5.3.3.1 Performance Metrics

To compare the performance of the proposed PPS approach, the following three performance metrics are used:

1) Average Network Delay (AND), measured in msec and is defined as the average amount of time required to deliver a data unit to the AP.

2) Average Packet Delivery (APD) is the average percentage of transmitted data packets that succeed in reaching the AP reflecting the effect of delay on data delivery over the utilized data delivery approach.

3) Average Network Lifetime (ANL) is a measurement of the total rounds the deployed network can stay operational for.

While studying these performance metrics, we vary five main parameters:

1) The size of the network in terms of total LN count. This reflects the application’s complexity and the scalability of the exploited routing scheme.

2) DC count while fixing the LN count.

3) Pause time for MDCs as a major delay factor.

4) Sensing field radios (R) per MDC.
5) Average packet generation rate per time round \((D)\) as an indicator of the traffic load across the network.

6) Price at the GW level to observe the influence of its increments on delay and lifetime.

### 5.3.3.2 Simulation Results

Figures 5.18 – 5.26 show our simulation results according to the aforementioned metrics and parameters while muting the effect of price on data delivery. However, the price influence is demonstrated in Figures 5.27 and 5.28.

In Figure 5.18, we see that the APD decreases for all three delivery schemes as the size of the network increases in terms of LN count. This is to be expected since a larger network size implies longer paths and higher probabilities for link loss. A larger node count raises the risk of node failure and, hence, dropped packets. Thus, choosing smaller peripheral networks is better for the quality gain according to Eq. 5.28. The results in Figure 5.18 show that PPS substantially outperforms both PODV and PRRA. Several factors contribute to this superior performance of PPS. First, our scheme assumes a heterogeneous environment with abundance of data sources where several copies of the required data may be available. The other schemes suffer from their own disadvantages. PODV, for the one hand, relays on multiple history-based routing tables which is inadequate for heavy mobile data traffic. PRRA, on the other hand, is a source-routing protocol, meaning route maintenance mechanism does not locally repair a broken link. In addition, the connection setup delay is higher than that in table-driven protocols. Even though the protocol performs well in static and low-mobility environments, the performance degrades rapidly with increasing mobility. Furthermore, PODV and PRRA are not aware of the link state in the cloud beyond their corresponding gateways.
Figure 5.18. Average packet delivery vs. LN count (100 GWs, 5-30 DCs).

Figure 5.19 examines APD rates with network size expressed in DC count. We note that while PODV and PRRA show a declining performance almost similar to that of Figure 5.18, PPS shows an improvement in delivery as the number of data collectors increases. This is because PPS’s architecture incorporates at its core the existence of DCs, which gives it an advantage over other delivery schemes. PODV performs slightly better than PRRA since, as we mentioned earlier, PRRA is challenged by increased mobility, which is subsequent to increases in DCs.
Figure 5.19. Average packet delivery vs. DC count (100 GWs, 4500 LNs).

Figure 5.20 compares APD against pause time. We see that for all three schemes, the delivery increases slightly as the pause time for mobile nodes increases. This is normal since more pause time ensures a wider window for control and data message exchange, especially for large data loads. Pause time also helps PRRA to discover unknown routes via flooding the network with route requests. However, PPS shows again a better performance compared to its rivals. At its worst case; PPS is 28% better in delivery than PRRA, which outperforms PODV as mobility decreases (i.e. pause time increases).
In Figure 5.21, we compare Average Network Delay (AND) with sensing field radius ($R$). We see that delay and size of sensing field are directly proportional. Yet, because PPS utilizes multiple data collectors per sensing field, in addition to adopting an approach that serves data-sensitivity by defining RNs and FTRNs (Algorithm 6), its delay increase is steadier and lower than that of its rivals.
The effect of utilizing data collectors and the FTRN approach (Algorithm 6) in PPS is further proven in Figure 5.22, where its Average Network Delay (AND) shows a sharp decline unattainable by neither PODV nor PRRA as the DC count increases.
Figure 5.22. Average network delay vs. DC counts (100 GWs, 4500 LNs).

The relation between AND and pause time is linear and directly related (Figure 5.23). The two delay components are correlated. But PPS shows an exceptionally better performance in terms of lower delay due to its delay-sensitive routing and source selection approach as explained in Algorithms 5 and 6.
Figure 5.23. Average network delay vs. pause time (100 GWs, 5-30 DCs, 4500 LNs).

In Figure 5.24, the increase in generation rate causes PRRA to increase its route discovery process which exhausts the ANL. In contrast PSS has a higher ANL because its data delivery scheme is top-down. LNs are not exhausted by routing loads. Rather, intermediary GWs are responsible for replying to data requests issued by APs. In addition, our categorization of LNs to relays and FTRNs has an apparent effect on reducing the transmission load over the collective set of LNs within each peripheral network.
Figure 5.24. Average network lifetime vs. $D$ (pkt/round) (100 GWs, 5-30 DCs, 4500 LNs).

Figure 5.25 shows that the increase in DC count significantly improves the lifetime of the system under PPS. More DCs facilitate delivery according to PPS algorithms and relieves LNs from further relaying tasks. As for PODV and PRRA, we note from Figures 5.24 and 5.25 that PRRA delivers better performance under low mobility conditions, whereas PODV outperforms PRRA when more MDCs are present.
In Figure 5.26, we study the effect of MDCs’ pause time on ANL, which is a measurement of the total rounds the deployed network can stay operational for. We note that PRRA performs the worst among the three schemes, which agrees with the aforementioned observation on the relation between high node mobility and PRRA.

As for the influence of data price set at the GW level, we assign this parameter to GWs in PODV and PRRA by calculating their average performance in terms of the parameters defined in Eq. 5.24 (i.e. relaying capacity, lifetime and delay). Then, the GW price is incremented by some specific value in each round (e.g. 1 price unit per 1 round of lifetime).
Figure 5.26. Average network lifetime vs. pause time (100 GWs, 5-30 DCs, 4500 LNs).

Since the selection of GWs is influenced by their announced prices according to our scheme, we apply price increments up to 90% on each GW and remark on its effect on reducing average network delay and increasing average network lifetime, respectively. Figure 5.27 shows that PPS provides lower delays as the GWs price increase. This is because PPS’s resource management algorithm is built on this price consideration. The result indicates that our pricing scheme is successful in providing improved delays yet for higher prices. PODV performs better than PRRA in this regard since PODV’s routing is essentially based on reducing hop-count on end-to-end links, indicating better handling of end-to-end delay on the LN-to-GW level.
Figure 5.27. Average delay vs. Price increment percentage (100 GWs, 5-30 DCs, 4500 LNs).

Figure 5.28 shows how the three algorithms react to price increments with respect to ANL per peripheral network. Here, as the client pays higher prices, more LNs (represented by GWs interfacing each peripheral network) will participate in the public sensing service with their data, which consequently entails more consumption of their energy resources. In other words, more lifetime (in terms of energy or battery power) can be provided for a higher price. Again, PPS proves to successfully utilize its pricing function in a manner that conserves lifetime and provides it as a resource for higher bidders only. Here, we note that PRRA performs better since its algorithm does not overwhelm every node in the network by periodical table updates and hence
dramatically degrading the nodes’ battery power, which is the case with PODV that performs worst in terms of lifetime even with steady increments on GW pricing.

![Graph showing average network lifetime vs. Price increment (100 GWs, 5-30 DCs, 4500 LNs).]

**Figure 5.28.** Average network lifetime vs. Price increment (100 GWs, 5-30 DCs, 4500 LNs).

### 5.4 Summary

In this Chapter, we introduce two schemes for priced data delivery in integrated IoT architectures. First, we introduce heuristics for monetary-based courier relaying (MCR) that governs packet relaying and price negotiation in RSNs. Our scheme incorporates a criticalness function involving both ID and sensory attributes of the packets; with respect to its spatial and temporal properties. The decision to forward the packet to a courier depends on the criticalness bound to it, in addition
to the courier’s charge with respect to a packet’s threshold price set by the SN. Hence, direct transmission to access points may be considered if packet criticalness is high, or if no feasible courier is available. We compare our RSN scheme to other dominant mobile Ad-hoc delivery schemes such as AODV and NRRA in terms of energy consumption and delivery rate. Our simulation results show that MCR performs superiorly under varying topology settings in terms of nodal count and mobility.

In Section 5.3, we introduce PPS; a priced public sensing framework for heterogeneous IoT setting. Our framework is based on a multi-tier architecture that caters for mixed data sources (sensors) in addition to both stationary and dynamic data collectors. The aforementioned components are collectively interfaced, as parts of their peripheral networks, with a data cloud. This interfacing is conducted by corresponding gateways. The delivery scheme introduced for this architecture implements algorithms that realize delay-sensitivity and quality constraints of the data. Moreover, we have provided a dynamic two-tier pricing scheme that caters, on the one hand for social welfare of the system by observing constraints on resources such as energy and transmission capacity, while applying, on the other hand an extensive utility function that directs data acquisition in a manner that serves the client’s demands in terms of delay, quality of service and value of the data. We finally provide simulation results showing the efficiency of our data delivery scheme compared to other WSN and mobile ad-hoc schemes with respect to network size, lifetime, end-to-end delay and packet delivery ratio.
Chapter 6

Conclusion

The concept of the Internet of Things (IoT) grew from a mere project to develop identification technologies to a worldwide paradigm that incorporates all objects that are able to interact directly with local neighbours within small separate networks. Defining a bounding framework and an architectural model for IoT is still an ongoing topic of investigation. Radio Frequency Identification (RFID) and Wireless Sensor Networks (WSNs) are widely considered to be key technologies characterizing IoT, forming an integrated model known as RFID-Sensor Networks (RSNs). RSNs represent a heterogeneous platform enabling an abundance of applications into the IoT context. The exploitation of this platform will result in more functional, scalable and cost-effective systems. However, most of the proposed integrated RSN architectures in the literature are application-specific and fail to realize the challenges presented by ultra-large-scale deployments in terms of cost-efficiency, interoperability and connectivity. Common integration approaches fail as well in efficiently utilizing the ubiquitously available components in today’s wireless topologies which we referred to here as Courier Nodes (CNs). Such couriers include handheld, devices, smartphones and transceivers onboard vehicles and public transit with deterministic or semi deterministic mobility traces. CNs are pivotal in facilitating a multitude of IoT functionalities.

In this thesis, we argue that an IoT setting is always characterized by the following: 1) The ability to identify, 2) Seamless integration, 3) Ubiquitous connectivity and 4) Delay-tolerance. The delay-tolerance characteristic is a direct result of intermittent connectivity, an attribute associated with concurrent disruption/partitioning caused by nodal mobility. We address these
characteristics by first proposing (in Chapter 3) an RSN integration approach along with an optimal placement solution to attain coverage and cost-efficiency. In Chapter 4 we investigate two data delivery approaches over the aforementioned RSN architecture after incorporating mobile couriers to its network model. In Chapter 5, we expand our investigation by proposing two pricing models for our RSN-based topology, with an emphasis on service-based applications and incorporating a detailed delay model and a pricing utility function.

Investigated placement, delivery and pricing schemes presented in this thesis are summarized in Section 6.1. Some future research directions are outlined in Section 6.2.

6.1 Summary

In Chapter 3, we introduced the RSN integration approach (SIWR) that we set as the basis for our data delivery schemes for IoT architectures. Our RSN network model integrates RFID readers and wireless relays together into Super Nodes (SNs) that represent the most complex component of our integration scheme in terms of functionality and cost. Other components in our network model include Light Nodes (LNs) and Base Stations (BSs). The LNs comprise simple RFID tags and sensors which are not involved in any advanced processing or relaying. The precise placement of SNs in our scheme guarantees its cost efficiency while addressing coverage constraints. Hence, we introduce an ILP formulation to accomplish optimal placement for SNs. When compared to other common RSN integrated schemes: tag-sensor (TS), reader-sensor (RS) and mixed architecture (MIX), our approach proves to be better in terms of cost efficiency for considerably larger topologies.

In Chapter 4, we expanded our (SIWR) network model to include ubiquitous mobile Courier Nodes (CNs) within IoT environments. CNs were instrumental in our two RSN data delivery schemes. The first (DIRSN) uses an ILP formulation to minimize the delivery delay by
minimizing the total path length from a SN towards the BS without overwhelming the integrated network. This is achieved by locating a courier set that maintains the shortest path from each SN to a BS while considering their varying node/link capacities and load balance. When compared to performances attained by other RSN integration architectures, our delay-based approach showed substantially better results in terms of average delay, average packet loss and average generation rate. The second delivery scheme (URIA) is connectivity-based and employs an algorithm based on an SDP that maximizes the formulated network connectivity by maximizing the second smallest eigenvalue $\lambda_2$ of a Laplacian matrix representing the graph of the topology. This SDP was executed over our RSN model and connectivity was achieved by utilizing the mobility of CNs rather than relying on passive recipients or relays. This approach was compared to our (DIRSN) delay-based delivery scheme and to a third approach (ORP) that similarly maximized the second smallest eigenvalue $\lambda_2$ by selecting linkers among stationary relays rather than mobile nodes. When connectivity constraints are tighter, URIA performs better than DIRSN and ORP in terms of average delay and packet loss.

In Chapter 5, we further expand our investigation by incorporating pricing policies into our delivery scheme. We provide two pricing models. The first (MCR) is delivery-based and allows couriers to charge SNs in return of relaying their packets to BSs if the former chose to conserve their transmission energy and rely on CNs instead of attempting direct transmission. This decision is based on a data criticalness function incorporated into our delivery algorithm. The second pricing model is designed for IoT-driven priced public sensing (PPS) applications. Here, a slightly different network model is proposed where LNs are all assumed to include any data generating node within its own peripheral small networks. These networks are interfaced to a data cloud via gateways (GWs). At the top of the hierarchy, data requests are assumed to be initiated
by human clients who run service-based applications. The clients pass their data requests to the cloud via Access Points (APs) that determine, on behalf of the clients, the parameters of the requested data in terms of quality metrics and set, in accordance to the service agreement, a threshold reservation price to this requested data. The GWs, on the other end of the cloud, receive the requests and reply to them after validating them against the data available at the sensors within each gateway’s coverage. Some GWs are able to communicate with mobile data collectors (MDCs) whose mobility adds to the delivery delay factor. Hence, we propose a detailed delay model that addresses the constraints on delay and provides delivery alternatives for delay-tolerant and delay-sensitive data respectively. As for the pricing model, it is a twofold approach. First, the GW sets a price to its data based on the abundance of resources and the social welfare of its peripheral network in terms of lifetime and energy. Along with this price, the GW will send to the client/requester its quality and delay parameters for associated to the type of the available data. The AP enters these values into a utility function that also includes a history-based trust function to decide which among the replying GWs is most worthy of approval. We compare our delivery scheme developed for PPS against two common mobile Ad-hoc routing protocols after modifying them to better suit the data cloud model. Our PPS approach significantly outperformed its rivals in terms of average delivery rate, delay and network lifetime.

6.2 Future Work

Several future research topics can be derived from our work thus far. In this section, we point out some of these topics.

1) In Chapter 3, our integration approach (SIWR) only considered simple tags and simple sensors as LNs existing as separate entities in the topology. The simulation results presented in Chapter 3 were based on comparing SIWR to other integrated architectures:
TS, RS and MIX. It would be interesting to investigate a more complicated form of SIWR where in addition to SNs (integrated readers and relays), some LNs appear in the integrated form of TS (tag-sensor), as well. This flexibility would open SIWR to a wider variety of IoT applications. Examining the impact of our optimal placement of SNs on the cost-efficiency of this modified SIWR and comparing it to other integrated architectures is an issue for further discussion.

2) In terms of DTN routing (discussed in Chapter 4), an interesting class of DTN delivery schemes is social-based routing [99] which may also be referred to as Pocket-Switched Networking (PSN). This class is particularly valuable in scenarios where human mobility traces are involved in the routing process. PSN builds on social connections, investigating links within a given community and utilizing nodes with high connectivity rankings. Mobile nodes in such a topology are more likely to rationally interact with each other, which adds a new dimension to the delivery scheme. This concept may prove useful in many applications under IoT.

3) The study of nodal localization strategies has enriched a variety of research disciplines related to wireless and mobile Ad-hoc networking. In terms of IoT and integrated RSN architectures, investigating localization schemes is strongly encouraged by the identification possibilities provided by both RFID tags and sensor nodes. Discovering possibilities in this direction would help generate better placement and delivery strategies for hybrid IoT architectures.

4) Context-awareness is an important feature in service-based architectures which we considered in our priced public sensing scheme in Chapter 5. The abundance of sensed data in IoT and its correlation between wireless entities has recently increased
significantly. Understanding the context of each entity in a given environment is non-trivial. From a context-awareness perspective, integrating RFID and WSNs would result in smarter objects that provide their sensed context with a unique identification. When such objects produce information contributing to the same context, they could interact as groups or clusters to fuse information messages under a single identification to be accumulatively forwarded to sink nodes or base stations.
References


