Decentralized Peer-to-Peer Data Dissemination in Wireless Sensor Networks

Ricardo Simon Carbajo[1], Ciarán Mc Goldrick

School of Computer Science and Statistics
Trinity College Dublin (Ireland)

Abstract

Next generation Wireless Sensor Networks will operate as self-regulated ad hoc networks of tiny devices that sense, actuate and communicate in a collaborative, autonomous and decentralised manner. This new context aligns with the paradigms of edge and fog computing where it is paramount to reliably distribute data among a large sub(set) of consumer/producer devices. This paper presents a communications architecture for scalable selective data dissemination in wireless sensor networks which is comprised of a decentralised data distribution layer that is tightly coupled to a reliable gradient-based routing protocol. The system supports mechanisms for selective data pushing and pulling within the sensornet where unstructured Peer-to-Peer content distribution concepts are utilised to fairly distribute pieces of data amongst an overlay of consumer and producer nodes. The system has been evaluated under a variety of network conditions and scenarios, both via simulation and real-world deployment, and it is shown to be reliable, scalable, and capable of distributing data in a fair and efficient manner across the network.

Keywords: Wireless Sensor Networks, Peer-to-Peer Networking, Decentralized P2P Data Distribution, Selective Data Dissemination, Gradient Routing

1. Introduction

Wireless Sensor Networks (WSNs) are a particular realisation of wireless technologies characterised by the use of small, low power, low data rate, wireless devices with sensing and actuating capabilities. WSNs enable the monitoring of environmental conditions and infrastructures in a pervasive manner. In effect, the range of applications of traditional wired sensor technologies has been extended through the use of small unobtrusive wireless devices which may also form ad hoc networks. However, research has not yet unlocked the full potential of the technology, as envisioned by many authors in the late 90's. Pervasive networks of co-operative tiny mobile wireless devices, capable of interacting with the environment, will be the next research challenge in ubiquitous computing and will contribute to accomplishing the vision of ambient intelligence [1]. In this regard, one of the main goals is to transparently connect WSNs to the Internet such that WSNs can also be globally and transparently inter-connected and easily accessible to any user in any network, including via mobile cellular networks. At the same the fog computing paradigm will be enabled where services can be hosted at the edge of the network thus enhancing performance, quality of service and even quality of experience.

For this purpose, wireless sensor networks will be required to operate as self-regulated networks of static and transient devices behaving as autonomous colonies - sensing, acting and communicating in a collaborative way. Pushing and pulling data from any point in the sensor network is necessary such that nodes receive up-to-date information on its surroundings for collaborative decision making. When a node needs to communicate to a distant area within the same WSN, or to another WSN, backbone networks, such as the Internet, should be employed when possible. For instance, nodes sensing and acting over a wide area might require to be aware of certain events and data produced at,

Email addresses: `carbajor@tcd.ie` (Ricardo Simon Carbajo), `ciaran.mcgoldrick@tcd.ie` (Ciarán Mc Goldrick)

[1]Corresponding Author. Address: School of Computer Science and Statistics, O'Reilly Institute, Trinity College, Dublin 2, Ireland.

say, 30 hops away. Backbone networks can be used to transparently push data closer to, or into the requesting area, via gateway nodes placed at any point in the network. However, when a gateway is not accessible, a dissemination process will be required to reliably push data towards disparate areas. Moreover, data needs to be replicated and disseminated so that the sensor network acts as an autonomous distributed storage medium preserving data to be uploaded to other nodes, or to the Internet, from different points in the network in a reliable, robust and efficient manner. In this regard, data needs to be distributed to multiple points in the WSN to reduce the risk of data being lost in the event of devices or entire areas failing or being disconnected. This is something which could occur due to the error-prone wireless medium, the limited energy of motes, and/or the unreliable physical conditions of the environment where motes can be deployed. For instance, consider a scenario where sensors are tracking and monitoring glaciers which might suddenly break into smaller ice blocks. Climate scientists will be interested in periodically retrieving data from those small separated ice blocks from different accessible points of the isolated network at a later stage. Selective dissemination and storage of data at multiple points will give the sensor network such degree of required autonomy when performing data collection. As another example, consider a set of robots working on a collective task where the hazardous nature of the environment and conditions represent a high risk for their safe operation, e.g. scenarios of disaster rescue and exploration where robots are subject to be destroyed by the unstable surrounding elements. The group of sensing and actuating robots should work as a self-organised decentralised network capable of cooperating and adapting to the changing conditions. For this purpose, the decentralised dissemination of data to a sub(set) of robots will not only avoid data loss but also enable decision making based on historical data, perhaps in order to rearrange the cooperative strategy. This also increases data integrity, makes the network fault tolerant and reduces the appearance of the hot spot and flash crowd problems. Nevertheless, nodes should decide whether to participate in the dissemination and storage process according to their capabilities or the status of their resources. To achieve such a level of data distribution, a reliable data dissemination protocol is required which distributes data replicas over the network to a set of nodes. This goal necessitates a communications architecture composed of a set of reliable and efficient protocols capable of providing fault tolerant decentralised data distribution in unreliable scalable wireless sensor networks.

This paper presents such a communications architecture, known as the TinyTorrents (TT) framework. TinyTorrents leverages existing Peer-to-Peer (P2P) content distribution concepts from wired and wireless networks. Previous work [2] presented the TinyTorrents infrastructure in its centralized version. This paper describes the decentralized version of the system which provides the substrate for the development of co-operative decision-making applications for scalable wireless sensor networks. The TinyTorrents framework has been designed to efficiently distribute data inside the sensor network in a decentralized and scalable manner. As a P2P architecture, TinyTorrents moves the resource management to the edge of the network, i.e. to the "peers". This is achieved by enabling direct communication among peers, which encourages them to cooperate autonomously amongst each other in managing the resource. This behaviour enhances the reliability and availability of the system, as they are more resistant to individual node failures. Peer autonomy also means that peers are equipped with discovery protocols, which allow them to find functioning peers in the network when neighbouring peers have failed. This ability for "self-organisation" greatly adds to the reliability of P2P systems, mainly when the network operates in a fully decentralised fashion through the use of structured or unstructured discovery mechanisms. TinyTorrents is designed as a two-tier architecture. The upper tier is a selective data dissemination protocol for reliable data distribution in sensor networks, i.e. the TinyTorrents protocol. This protocol employs P2P communication mechanisms to replicate data amongst nodes in a fair, efficient, reliable and scalable manner across the WSN. The TinyTorrents protocol operates above a suitable routing mechanism. For this purpose, a gradient-based reactive routing protocol, known as Ubiquitous Mobile Gradient (UMG), has been designed to achieve efficient P2P communication. This incorporates an efficient lookup mechanism to support content distribution and discovery techniques in TinyTorrents. The TinyTorrents system has been evaluated by simulation and in a real testbed comprised of 64 TelosB devices.

The rest of the paper is organised as follows. Section 2 explains the rationale for the concepts and approaches presented in this paper and positions the work in the context of the state of the art in the areas of data dissemination and structured and unstructured P2P systems in wireless sensor networks. In the following section 3, the TinyTorrents system architecture is presented. The UMG routing protocol [3] is also briefly described in section 4. Subsequently, the TinyTorrents protocol is expounded in section 5. Section 6 presents a performance analysis of the system through both simulations and real world deployment, and assesses its performance when compared against epidemic dissemination protocols. Section 7 summarisses the work, its findings and its contributions to the domain.

2. Rationale and Related Work

Data dissemination techniques over wireless sensor networks have been proposed to reliably distribute data with the aim of achieving high data delivery rates and low delivery latency whilst minimizing communication and energy consumption. Much of the research in wireless sensor networks targets the problem of data dissemination from the perspective of the initiator of the communication: i) source-to-sink, i.e. pushing or ii) sink-to-node i.e. pulling. A sink node pulls data by sending a request to the source node. However, a source node may decide to push data to the sink at any given time. Pushing data towards a sink, or set of sinks, can be seen as a data collection process. Sinks can be placed at different points in the network for continuous collection or can act as opportunistic or nomadic points of collection which move to another area once data has been exchanged. Nomadic mobile sinks are sometimes known as data mules. Sink nodes usually have more resources and can communicate with other networks such as the Internet. Hence they are employed as gateways to perform pulling activities over the WSN and to push data outside the WSN. For reconfigurable systems, pushing data into the sensor network is employed for programming purposes where the goal is to reliably deliver big chunks of data to all, or most of the nodes, in the network.

Initial approaches in the area of dissemination employed the flooding and gossiping mechanisms. However, flooding and gossiping, in their basic forms, are considered low complexity, inefficient solutions which do not optimise the process of data dissemination. One of the first approaches in the design of data dissemination mechanisms for wireless sensor networks was a protocol called Directed Diffusion [4]. In Directed Diffusion, source nodes describe data using attributes, introducing the concept of data-centric routing (replacement for the tradition address-centric approach). Around the same time, the SPIN [5, 6] protocol tackled the problem of data dissemination from a data-centric perspective with a negotiation-based paradigm in which nodes decide whether to acquire the data from neighbour nodes. These two protocols establish the concept of metadata packets, known as "interests", which travel the network to inform other nodes of the availability of data while preparing the path for data communication. In this regard, data dissemination protocols, where the sink propagates the interest and the source responds with data, might be classified as Sink Oriented Data Dissemination protocols. Examples of this type are Directed Diffusion [7, 4, 8], TTDD [9], Declarative Routing Protocol (DRP) [10] and GRAB [11]. When sinks become mobile, these algorithms need to update their sources of information. For instance in [12], a data dissemination algorithm employs the concept of pseudo-distance to create an estimation of how far the mobile sink is as it travels through the network. This metric is calculated based on the concept of forwarding interest packets which the mobile sink node broadcasts periodically for other sensor nodes to update or estimate the nodes new position. The approach assumes that all links are bidirectional and no control messages are lost and seeks to reduce the control overhead involved in forwarding data to mobile sink nodes. The design of Sink Oriented Data Dissemination protocols revolves around the subscribe-publish paradigm where nodes disseminate the interest or query towards potential sources of data such that data is forwarded to nodes which manifest an interest [13]. On the other hand, Source Oriented Data Dissemination protocols are characterised by source nodes initiating the dissemination of metadata towards the sink. On reception, the sink performs some data acquisition process which might be based on negotiations. A good example of this is SPIN [5]. The Source Oriented Data Dissemination protocols are based only on source nodes publishing and disseminating metadata towards the network in order to alert nodes on the availability of data. This type of protocol can easily be enhanced with a subscribe mechanism if the dissemination process can be controlled. The adoption of each scheme brings different benefits and drawbacks related to scalability, network topology dynamics, resource-efficiency and real-time data acquisition.

Dissemination protocols have also been employed for multi-hop network reprogramming where large objects of data/code are disseminated over the network in a reliable manner. Examples of these types of protocols are Pump Slowly Fetch Quickly (PSFQ) [14], GARUDA [15] and Reliable Multi-Segment Transport (RMST) [16] which employ hop-by-hop retransmissions for reliability. Multihop Over-the-Air Programming (MOAP) [17] is another approach to network reprogramming over multihop networks which delivers the whole object to a node before the latter become a source for the next hop. In [18], the Multihop Network Reprogramming protocol (MNP) adds a sender selection algorithm which attempts to guarantee that at most one source is transmitting within a neighbourhood to minimize collisions. Other approaches rely on network coordination to minimize collision at a higher complexity cost. Splash [19], a dissemination protocol for large objects in wireless sensor networks, creates a tree structure rooted at the source and combines recent advances in constructive interference broadcast and multiple-channel pipelining to eliminate contention overhead. It controls when and which set of nodes transmit packets based on their height to root, thus reducing contention. Recent works address the problem of dissemination for reprogramming wireless sensor networks in a fast and energy efficient manner by i) using a correlated tree structure which is constructed according to the wireless link qualities and correlations [20] and ii) reducing the transmit power in a multi-hop wireless network using game theory, where every node is modeled as a player in a game model and the action of forwarding the packet depends on the actions of other nodes [21]. Specifically in TinyOS [22], the embedded operating system employed in the evaluation of this work, two data dissemination protocols are available for reprogramming purposes: i) Deluge [23], and ii) Typhon [24]. However, for small to medium data size dissemination, DIP [25] and DHV [26] are employed in TinyOS as lightweight solutions. They operate by finding nodes in need of data updates in order to maintain consistency over the network and employ the Trickle [27] protocol for maintaining code updates. Trickle utilizes beacons to disseminate data which is smaller than the payload of a packet, and self-regulates the periodicity of the update beacon based on the changes in the neighbourhood.

All of these dissemination protocols are employed to deliver the whole object to a node before the latter becomes a source for the next hop. Pushing data to the network is achieved by these protocols in a reliable manner. However, they are based on hop-by-hop data communication primitives, forwarding data packets through pre-established routes or in an epidemic fashion. These algorithms become inefficient in terms of communication when a dynamic subset of nodes in a sparse network is comprised of producers and consumers of data with the remainder of the nodes acting as relays. This motivates the research problem of how to disseminate data reliably and efficiently amongst a subset of consumer and producer nodes in scalable and unstructured networks.

The novel approach presented in this paper provides selective data dissemination to a sub(set) of consumers in the network in a collaborative manner which balances the traffic load. This is achieved by leveraging the concept of peer-to-peer (P2P) networking which enables distribution of the burden of centralised servers by turning networks into a set of self-organised devices capable of communicating with each other and sharing resources. This paradigm offers advantages such as i) network traffic load distribution, ii) fault-tolerance, iii) enhanced reliability, and iv) robustness to change. Decentralised P2P approaches scale better and can adapt to local changes in the network topology, and are ideal for data dissemination in WSNs. However data discovery poses a challenging problem in decentralised P2P networks. In the area of ad hoc wireless networks, data discovery presents greater challenges than in traditional wired networks. This is mainly due to the unreliable wireless medium and the inefficiency of routing to distant nodes. Both structured and unstructured mechanisms have been designed for data discovery in wireless ad hoc networks.

2.1. *Structured P2P systems*

Structured P2P systems create virtual overlays employing Distributed Hash Table (DHT) schemes for data search and retrieval through the network. A variety of DHT-based schemes has been employed in WSNs for data management [28] and for developing routing protocols capable of scaling [29, 30]. One of the first approaches to distribute, store and query data in WSNs was published in the papers Data-Centric Storage in Sensornets (DCS) [31, 32] and Geographic Hash Tables for Data-Centric Storage (GHT) [33]. The authors presented the idea of distributing data by replicating it amongst nodes in the network such that a failure in the area will not affect the already gathered data. In order to distribute the data, a geographical routing protocol called Greedy Perimeter Stateless Routing (GPSR) [34] is employed. Another approach was proposed in Chord for Sensor Networks (CSN) [35] as a protocol which follows the principles of the DHT-based Chord algorithm [36] to provide a lookup mechanism capable of searching data with logarithmic complexity. The authors claim it scales well as it maintains a finger table of $O(\log(n))$ entries and employs $O(\log(n))$ messages to locate data. Cluster hierarchies and overlays are formed such that virtual neighbours are closer geographically. The same authors propose a case for P2P overlay networks in sensor networks [37]. This paper outlines the design goals to consider when applying overlays in sensor networks, proposing a DHT-based protocol known as Tiered Chord (TChord). A solution for implementing a DHT in MANETS was proposed in Ekta [38]. The idea provides a peer-to-peer substrate by the integration of DHT-abstraction Pastry [39] in the network layer, employing DSR [40], a reactive routing protocol for multi-hop communication. Ekta updates its routing table and leaf set by snooping packets. A similar approach to Ekta, that integrates Pastry with the AODV [41] routing protocol, was later presented as MADPastry [42]. DHT tables are formed considering physical locality. It employs a set of random landmark keys spread evenly in terms of geographic coordinates which define temporal cluster nodes. In a paper by Zheng et al. [43], a DST is used to maintain the ranges of keys in a DHT, therefore obtaining the range query, i.e. all keys within a range, and the cover query, i.e. all ranges for a key. Ghose et al. proposed a DHT approach with circular virtual overlays over a geographical routing protocol, the Greedy Perimeter Stateless Routing (GPSR) [34], which finds optimal routes by using local information at every hop [44].

There has also been research on constructing DHT according to the topology of the network, such that virtual neighbours in the overlay are also close in terms of routing (hops) [45]. For instance, Topology-based DHT (T-DHT) [46] selects “reference” nodes by flooding mechanisms to offer support for other nodes to perform triangulation and to calculate the relative position of a node. Then, a virtual two-dimensional coordinate system is generated and mapped into a DHT which is used to store and retrieve data deterministically. A similar approach for data-centric storage and routing is also proposed in GEM [47], which uses graph embedding and tree routing, maintaining a two hop neighbourhood. A routing protocol which applies DHT concepts on top of the link layer was presented as Virtual Ring Routing (VRR) [48]. The goals in designing the routing protocol were: i) to avoid flooding and ii) to avoid location-dependent addresses. On the other hand, scalability issues were not evaluated. Routing between virtual neighbours in the Pastry-based overlay ring is performed in such a way that there is knowledge about other nodes in the network, rather than only predecessors and successors. The complexity (in number of hops) to locate a virtual node can be reduced from $O(\log(n))$ to $O(\sqrt{n})$. This is achieved by taking into account link layer information of closer nodes when routing in the virtual ring. In the same line, Awad et al. presented Virtual Cord Protocol (VCP) as a DHT-based routing protocol which sits on top of the MAC layer [49, 50]. It also takes into account geographical proximity (2 hops away) of virtual nodes to minimize communication. Like in VRR, routing is based on virtual and routing knowledge. Finally, ScatterPastry [51] is another DHT approach using Pastry on top, or integrated, at routing level. It is evaluated in a real WSN testbed called ScatterWeb. The authors try to minimize energy consumption and address scalability by using Pastry concepts both on top of, and in combination with, the Destination-Sequenced Distance Vector (DSDV) [52] routing protocol. In trying to avoid mismatch between the structured overlay and the physical network, the authors in [53] create a 3-dimensional overlay for P2P over MANETs which logically interprets the physical relationship of a peer with its (logical) neighbour peers. Each peer constructs a 3D rectangular coordinate system with three planes that divide the space into six dimensions and eight octants, where the peer acts as the origin of the structure and each neighboring peer obtains a transient logical identifier that reflects their physical proximity in the overlay. Weights are assigned to each link using the inverse distance function, providing connectivity to a nodes neighboring peers on the basis of their hop distances which are obtained from the underlying routing protocol, OLSR [54]. Periodic probe messages between direct logical neighboring peers are employed to maintain the overlay. More recently, the Motion-MiX routing protocol [55] has been designed to track mobile nodes in a wireless network in time and space, focusing on their data resources. The hashing of the node identifier is employed to generate a logical address interval (LAIs) in the DHT for the data resource and each node can take several LAIs. In a one-hop communication manner, the network stores the motions of every node and the LAIs as distributed encounter milestones in a cooperative manner, and thus provides a shared virtual geographic map to approximate the motion trails of every node and its resources. However, as this algorithm needs to exploit the mobility assisted milestone sharing, it thus depends on the motion of nodes.

Many of the above structured P2P data discovery concepts explore the impact that unreliable peers produce when entering and leaving the structure of the DHT. Solutions have been proposed to ameliorate this effect by considering peer proximity thus enhancing DHT robustness. However, when facing lossy networks where peers suffer from short and long periods of connectivity disruption, and may even have mobility, DHT lookup schemes have proven to be ineffective. On the other hand, unstructured discovery systems have also been proposed which do not require the formation and maintenance of overlay networks. In Wireless Sensor Networks, unstructured lookup mechanisms offer a good solution for data searching in dynamic network topologies where structured lookup mechanisms (DHT-based) lose performance. The benefits of unstructured query systems are highlighted in scenarios with a high degree of data replication over the network. This situation makes the unstructured searching process efficient when compared to DHT-based methods at a much lower complexity and low maintenance cost.

2.2. Unstructured P2P discovery mechanisms

These mechanisms can be classified in two categories: i) Flooding and ii) Random Walks (RW). In Flooding, three types of enhancements can be identified [56]: i) Expanding Ring Search (ERS), Blocking Expanding Ring Search (BERS), and iii) Local Indices (LI). Random walks [57] are subdivided into a) K-Parallel and b) Agent Cloning. In the area of WSNs, different approaches exist which employ the above methods as the basis for unstructured discovery. For instance, the comb-needle mechanism [58] is an unstructured approach where the searching process benefits from a previous data dissemination phase. This paradigm exploits the redundancy factor of information spread over the network where queries do not need to reach the node/area where the event is produced, but rather locate a near node which can provide either the data itself or information about the location of the data being queried. This can also be categorized as a biased k-parallel random walk technique. Dimakis et al. proposed an unstructured solution on how to enable ubiquitous access to distributed data in WSN [59, 60]. The goal is to retrieve a set of distributed data packets by querying as many nodes as the number of data packets required. Storage nodes are randomly selected and data is pre-routed with a complexity per data node of $O(\ln(n))$. Distributed erasure codes are employed as a solution to achieve reliable distributed storage. Similarly, Rachuri et al. presented Increasing Ray Search (IRS) [61], a highly scalable, density independent unstructured solution to achieve efficient search in high density scenarios when compared to ERS and Random Walks. This paper aims to minimize the number of messages employed in the query mechanisms in an unstructured network in a non-deterministic way. In ACQUIRE [62], a packet is injected containing an active query which employs random walks, biased random walks or even a predetermined path. Each node receiving a query packet performs, if information is not available, an on-demand request for information, within a scoped distance in terms of hops, in order to improve on the selection of the next neighbour. When the active query is completely resolved, the query reply is sent directly to the source. A combination of biased random walks for Pull-type unstructured search where sensor nodes are considered stationary is presented in [63]. Sensor nodes need to be aware of their neighbourhood and the distance in hops to the sink, i.e. the level. A set of protocols are presented which employs a combination of Several Short Random Walks (SSRW) and Level Biased Walks (LBW). In LBW a query is propagated from the sink in such a way that it is forwarded to a neighbour with a higher level until it reaches a limit. If the query is not successful, another is started. A query walk can alternate a sequence of steps using LBW and then perform SSRW. The idea of LBW is to increase the coverage by reducing the correlation of visited nodes. The protocols are efficient in terms of energy, communications and latency when compared to simple random walks and multiple random walks techniques. In the category of biased random walks, searching based on Bloom Filters (BFs) [64] come as a probabilistic-based memory-efficient solution. The Bloom Filter is a memory efficient structure for data compression which has been used as a query mechanism in peer-to-peer networks in the Internet. The filter acts as a probabilistic membership structure in the form of a bit vector which can be queried to find out, with a certain likelihood, whether an element has been previously stored. Different approaches to BF-based searching exist. The simple one is that where the BF storing a description of the content in a node is spread to all the nodes in the network. Queries are then resolved by checking the BF table and the use of a routing protocol. On the other hand, BFs have been employed as data-centric routing mechanisms [65, 66, 67] where queries are forwarded based on the BF information cached at a given node about its neighbours. The routing process is usually performed by a greedy approximation which selects the next node whose BF best matches the BF in the query. Approaches such as the Attenuated Bloom Filter (ABF) [68] aggregates filters according to their proximity in terms of hops while the Exponential Decay Bloom Filer (EDBF) [69] applies a decay proportional to the hop count to introduce noise in the received BF. In the area of WSN, Hebden et al. [66] employed a hierarchical BF structure where a cluster formation protocol assigns nodes to monitor an area via caching data from packets in a counting Bloom filter - a BF which keeps count on how many times an element has been inserted. Query routing is performed between cluster heads according to the BF structures. Li et al. adopted a flat-network approach presenting the Scope Decay Bloom Filter (SDBF) as a structure to disseminate event hints [67]. Bloom Filter-based searching has been proved to increase the query success rate, reduce energy consumption and decrease latency when compared to flooding-based or random walks at the cost of BF dissemination overhead. Resource discovery mechanisms also seek to exploit information from the dissemination process at each local peer. For instance, each peer in [70] contains a database with local information of resources and routing information and is leveraged to improve the accuracy during discovery. A ranking function on this data helps prioritize the information transmitted according to the interest and mobility of peers.

Moreover, a set of algorithms in unstructured data dissemination employ mobile nodes to spread the packets and update the network view of available resources. This mechanism depends on the mobility of nodes for the dissemination activity. For instance, the Energy-Efficient Message Dissemination protocol (EMD) [71] performs data dissemination in delay-tolerant WSANs. The algorithm seeks to disseminate data from mobile actors as they come in contact with static nodes which receive and store the messages, and subsequently forward them to other participants when they come into communication range. This is performed in an energy-efficient manner and within a given delay constraint and coverage rate. Similarly, in “A spatio-temporal approach to selective data dissemination in mobile peer-to-peer networks” [72] mobile nodes, when encountering other nodes within communication range, enter into an exchange of so called reports which contain information on the age and distance of the data resource. By ranking the set of cached reports from previous mobile nodes encounters, based on the aforementioned features, only a subset of reports is transmitted according to the novelty probability of each report being new to the other node and the bandwidth and energy of each moving object. The algorithm assumes that each moving object knows its current location through some localization technique, such as a GPS. In a further example, TOSS [73], a token-passing, multi-point relays, data dissemination scheme in mobile Peer-to-Peer environments, employs a location-based scheduling approach to disseminate messages. These data dissemination algorithms improve their performance by using location information at the cost of energy demanding techniques and devices.

In this paper, unstructured searching paradigm concepts are employed in the design of the TinyTorrents system which employs a push phase to spread information on the available data and a pull phase which leverages the spread information for the efficient P2P data distribution process. In addition, the TinyTorrents unstructured P2P layer makes use of a gradient-based routing protocol, known as Ubiquitous Mobile Gradient (UMG) [3], which employs Bloom filters for data advertisement and discovery. Other approaches in the literature have employed unstructured P2P data distribution on top of routing protocols. For instance, in “An Efficient Overlay for Unstructured P2P File Sharing over MANET using Underlying Cluster-based Routing” [74], an unstructured approach leverages the underlying cluster-based routing protocol (CBRP) [75]. Cluster heads need to be elected and a proactive paradigm is employed for updating peers in the cluster - our proposed UMG routing protocol operates in an on-demand fashion and cluster heads are not required. In [74] data packets and other unicast packets are forwarded using the GPSR algorithm [34] and thus requires location-based techniques which consume energy. Each peer maintains a peer-routing table that stores the information of the root-peer and neighbor peers to build up a minimum-spanning tree which removes distant redundant links and constructs an overlay closer to the physical network using location information. However, the algorithm is not fault tolerant as it relies on one of the peers in the P2P network to act as a root-peer to connect all peers.

The TinyTorrents system proposed in this paper operates in a decentralised manner and does not need to maintain a logical overlay network to provide selective data distribution over the sensornet to consumers. The TinyTorrents protocol provides a fault-tolerance solution which eliminates the unique central tracker and makes the system scalable. The decentralised design is based on the concept of every consumer node (peer involved in the data distribution process) acting as an opportunistic unstructured local point of coordination/information (i.e. partial tracker which keeps track of a subset of the peers in the swarm within a nearby scope. The scope is not a fixed value and depends on the distribution of consumer nodes in the network. The Ubiquitous Mobile Gradient (UMG) routing protocol has been designed to reliably transport data packets within the local scope of each consumer, providing key functionality to make the system scalable and decentralized. UMG operates on a fully reactive manner and does not require localization information, thus being energy efficient, specially as communication in the network only occurs when data needs to be produced or consumed.

3. The TinyTorrents System Architecture

The concept of TinyTorrents leverages existing P2P content distribution ideas from traditional wired networks to replicate and store data within a Wireless Sensor Network and on the Internet. It provides mechanisms to distribute data in such a way that the WSN can be converted into an autonomous decision-making system capable of operating independently and also interoperating with other networks via the Internet. The TinyTorrents framework comes from the idea of employing both existing technology and concepts from the BitTorrent [76] protocol to disseminate WSN data in a peer-to-peer fashion, having each sensor node acting as a logical peer in the Internet BitTorrent network. Hence, the TinyTorrents framework can be divided in two functional entities which transparently integrate with each other: i) the Gateway-BitTorrent Communications Architecture, and ii) the WSN Communications Architecture.

3.1. The Gateway-BitTorrent Communications Architecture

The TinyTorrents framework provides a generic and versatile way of distributing data from the WSN to the Internet, interconnecting the networks in a transparent manner. The interconnection between the Internet and the WSN employs existing peer-to-peer (P2P) ideologies where data is encapsulated into torrents and stored at different nodes in the Internet. This is achieved via the use of the BitTorrent protocol to create a distributed fault-tolerant database. For BitTorrent connectivity purposes, the Vuze P2P content distribution client-server application [77] is employed. The interconnection between Vuze and the WSN has been achieved through the development of the “Vuze TT Plugin”, depicted in Figure 1. The plugin deals with translating WSN torrents into BitTorrent format and making the data accessible in a variety of ways.

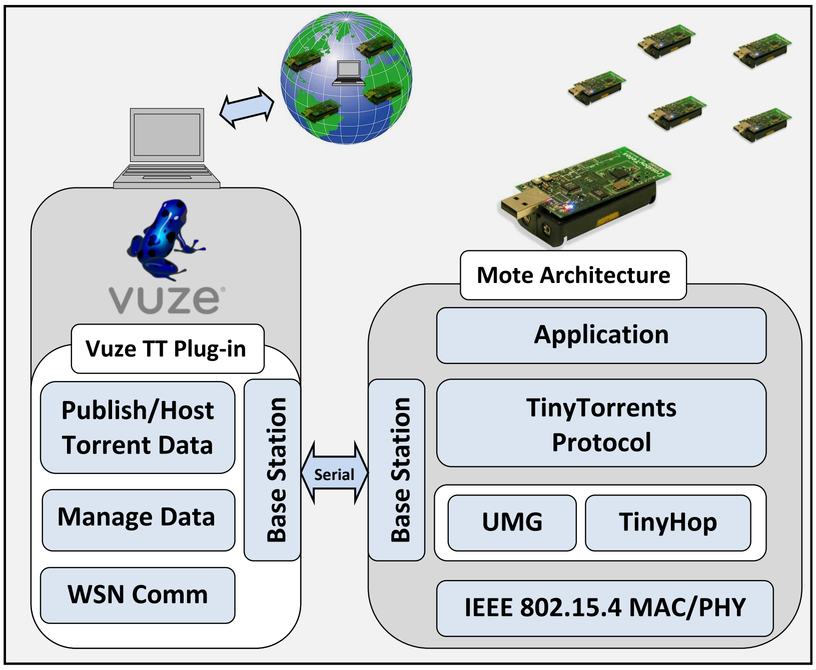

Figure 1: The TinyTorrents Communications Architecture

3.2. The WSN Communications Architecture

A communications architecture runs in each node of the Wireless Sensor Network capable of providing selective data distribution over the sensornet. The architecture provides functionality for each node to connect to the "Vuze TT Plugin" and thus act as a gateway. The architecture for each sensor node (mote) is depicted in Figure 1 under the heading "Mote Architecture". A data distribution protocol has been specifically designed to suit the constraints of WSN, known as the TinyTorrents protocol. The TinyTorrents protocol employs P2P data distribution techniques to replicate data amongst peers in a fair, selective, efficient and reliable manner across the WSN. The protocol, which employs concepts from the BitTorrent protocol, makes use of the metadata files to describe data, and generates unique keys to identify data files. Similar to the BitTorrent protocol and other content distribution protocols, files are partitioned into pieces which are then distributed amongst other peers which, in turn, become distributors. This approach suits WSN where selective data replication is required as a mechanism to protect against network failures and to push data closer to consumers. In addition, the concept of every peer being able to become a distributor ameliorates the burden at the data source node and balances the network traffic to help avoid partitions. Two increasingly complex designs of the "Mote Architecture" have been developed. Initially, in version 1, the TinyTorrents protocol operates with a single tracker, a central node which manages the list of peers containing the data, i.e. centralised version. The results of this work are summarised in [2]. However, this paper focuses on the decentralised version of the TinyTorrents protocol which has been designed to operate in a fully decentralised manner where the single point of failure, the tracker, is eliminated. Instead, every consumer (peer) becomes a partial tracker for the data it holds. Unstructured discovery mechanisms are proposed to discover proximate peers holding pieces of data. The approach is efficient in terms of communication and does not require constant maintenance or updates. In addition, efficient peer-piece selection mechanisms are presented which distribute the traffic load in the network in a fair manner while fostering diversity and reducing communications. The TinyTorrents protocol manages the logical distribution of data and thus needs to sit on top of a multi-hop routing protocol for the end-to-end multi-hop connectivity. Two routing protocols, Ubiquitous Mobile Gradient (UMG) and TinyHop, have been designed to address the constraints of WSN while offering cross-layer support to the TinyTorrents protocol. The first routing protocol, employed in version 1 of the "Mote Architecture", is called TinyHop [78], which is an on-demand, peer-to-peer, flat-topology, reliable routing protocol. The second protocol is called Ubiquitous Mobile Gradient (UMG) and introduces improvements with respect to TinyHop by using the concept of gradients while providing key functionality to make the system scalable and decentralized. UMG also introduces an efficient mechanism to support lookup content distribution techniques in TinyTorrents and is designed to work in an environment where sinks can be mobile. The remainder of this paper focuses on the design of the protocols which shape the "Mote Architecture", specifically the UMG routing protocol (see Section 4) and the TinyTorrents protocol (see Section 5).

4. The UMG Routing Protocol

A dynamic gradient-based reactive routing protocol, known as Ubiquitous Mobile Gradient (UMG), has been designed for Wireless Sensor Networks [3]. UMG has been designed to operate as the main routing substrate for the TinyTorrents protocol, providing end-to-end communication between consumers and producers of data. UMG follows the reactive paradigm, employing eavesdropping to update neighbours rather than using periodic updates. It employs the gradient concept while offering reliable mechanisms for the creation, update and navigation of the gradient field. It supports point-to-point, multipoint-to-point and point-to-multipoint communication. UMG is address-centric in the sense that node addresses are employed for routing, but it also integrates support for data-centric routing by making use of Bloom filters [64] as compressed mechanisms for storing data descriptions. Bloom filter-based descriptors are integrated in the gradient formation thus providing a mechanism for nodes to advertise their services. The routing protocol provides functionality to efficiently search and compare descriptors. In this sense, UMG acts as a service advertisement protocol and facilitates the data-centric searching process. The possibility for each node to be able to describe itself allows for applications to create multiple overlays according to node's services or node's interests in data.

4.1. Gradient Setup

The Ubiquitous Mobile Gradient (UMG) routing protocol is based on the idea that a node wishing to be contacted must spread its gradient first. The proper formation of the gradient is a key element in the quality of the routing process, i.e. avoid local minimum points and inefficient path lengths. The node spreads its gradient either because has taken a producer/consumer role in the network or because another node requests to contact the node. A node spreading its gradient at any given time will be known as a "sink". Multiple sinks might exist in the UMG network which establish end-to-end communication. The rest of the nodes act as pure routers, i.e. relays for packets.

A node spreads (setup) its gradient by employing a reactive controlled flooding process which can be limited in scope by setting a maximum coverage distance in terms of hops. When flooding the network, the gradient origin node broadcasts a gradient message with its address and a descriptor; the descriptor contains the services/data which the node either provides or is interested in. Only the first gradient packet is forwarded from those received for a given gradient process from a gradient origin node. UMG implements a backoff mechanism for the proper formation of the gradient, which delays the next broadcast based on the hop distance from the gradient origin node. UMG employs the fastest route metric, i.e. the sender of the first packet received by a router node becomes the next hop when descending the gradient route back to the gradient origin node. This metric produces short path lengths (despite not always being optimal) while requiring low complexity and inherently considering the current congestion status of the nodes and the wireless medium. The key idea is that a node controls the number of packets to be broadcast such that the implosion problem is minimized. For this purpose, and in order to avoid loops, duplicate packets are detected. UMG adds a delayed extra retransmission (broadcast) of the first gradient packet in order to increase the reliability of the gradient formation when contention problems or inactivity of the node are given.

4.2. Gradient-based Data Transport

Once initial gradients are established, communication with the gradient origin node (sink) can be started from any other participating node in the gradient. This is achieved by descending the gradient in a reliable manner. Every data packet descending the gradient is acknowledged at each hop, either with an explicit packet or by snooping the next hop transmission. If there is node symmetry in the link, a local repair mechanism is launched which looks for a valid candidate to keep on descending the gradient. If the local repair fails, the entry in the routing table is provisionally disabled. The next end-to-end message will launch another local repair process without considering nodes with disabled entries for the gradient. This mechanism can be seen as a special one hop backtracking in which the failing node is not selected in the next end-to-end gradient descending process.

UMG provides end-to-end acknowledgements to enhance the reliability of the data delivery process. End-to-end acknowledgement packets climb the gradient towards the originator of the communication. For this purpose, UMG implements a short time-out cache structure which stores key information from received and sent packets in order to avoid cycles and identify whether the node forwarded a previous data packet. According to this, the acknowledgement packet is only broadcast by those nodes which previously participated in the sending of the data packet. If bidirectional end-to-end communication can not be achieved, the option of sending the acknowledgement packet via the gradient of the originator of the communication is enabled; for this purpose, the originator of the communication must have spread its gradient. In the worse case scenario, the originator node has the option to spread its gradient in requesting mode, i.e. requesting the gradient origin node to spread its gradient.

4.3. Mobility Support

The UMG routing protocol has been designed to support dynamic changing topologies, where sink nodes leave their current neighbourhoods to become part of other areas. The UMG routing protocol employs a probabilistic mobility assessment mechanism developed to estimate whether a node has changed neighbourhood [79]. A predefined table model utilises the neighbourhood activity information stored in a structure of temporal shifting Bloom filters to determine if the node is changing its position. When mobility is detected, UMG estimates the number of new and old neighbours and spreads the gradient with a limited scope in order to reach old neighbours and reconfigure them to forward data packets towards the new position.

4.4. UMG and the TinyTorrents protocol

UMG's design inherently supports the unstructured creation of clusters where some consumer/producer nodes might spread their gradient with a limited scope within a virtual cluster. Specially selected nodes can spread the gradient with a wider scope such that they act as an inter-area backbone network node. While UMG transports data within the scope of the gradients, the TinyTorrents protocol employs a selective data dissemination approach to push data to distant nodes in the network. In this way, scalability is achieved in a reliable cross-layer fashion. The rationale behind this approach is that nodes in large sensor networks tend to limit communication within a certain scope for efficient routing, as routing at higher hop distances has proved to be inefficient and impractical. In addition, the TinyTorrents protocol has been designed to advertise its services via descriptions of data. In this context, UMG's advertisement mechanism provides support for describing the "interest" of data producer and consumer nodes. Moreover, descriptors can be used for searching purposes in content distributed algorithms. This can be exploited by higher layers in areas like distributed data fusion, service discovery, distributed storage or swarm decision making.

5. The TinyTorrents Protocol

The TinyTorrents protocol offers an efficient mechanism for data publication and selective data distribution over the sensor network which allow for a collaborative distribution of data by employing application-level decision policies. TinyTorrents provides service advertisement and discovery mechanisms which allow for data tagging using human readable vocabulary. A data file is represented with a file called "TinyTorrent" also referred as "torrent". A torrent contains a description of the data and some data control information, for instance the number of pieces in which the data is divided, data integrity values, and the next node to contact in the discovery of other peers, i.e. the tracker node. Peer-to-peer data communication is achieved amongst nodes belonging to the swarm of the torrent, seeking to balance resource consumption and to dissipate the burden of data transfers and network overhead fairly.

5.1. Phases of the Protocol

The TinyTorrents protocol is comprised of the following phases which are executed when a node in the network needs to publish or to acquire some data file:

5.1.1. Publish Phase

This phase is initiated when a node, known as the “initial seeder”, contains data to be published, i.e. the node is a “producer” in the network. A torrent is created which describes and represents the data file in the network. Every node in the network potentially participates in the dissemination of the torrent file. Different dissemination strategies can be employed to distribute the torrent. When a torrent is received at a node, the application layer decides whether to acquire the data file associated with the torrent and whether to forward the torrent file. Nodes starting to fetch the data file for a received torrent become “peers” of the swarm of the torrent.

5.1.2. Peer List Request Phase

A node wishing to fetch the data file associated with a received torrent, thus becoming a “consumer” of the torrent, needs to request a list of peers, known as the “peer list”, by contacting a designated tracker node. A tracker node maintains a list of peers participating in the data fetching process of each torrent. At any given time, the complete list of peers for a torrent is known as the “swarm of the torrent”. The tracker is in charge of selecting a small subset of the swarm of peers to send back to the requesting node, the latter automatically becoming a peer in the swarm of the torrent.

5.1.3. Handshake Phase

Once a node receives a list of peers from which to acquire the data file, a handshake process is performed with each of the peers in the list. The handshake process is responsible for finding out: i) whether the peer is still contactable and, ii) the list of pieces of the data file contained by each peer. The data file is segmented into smaller “pieces” of data which become the atomic data unit in the TinyTorrents protocol; a piece of data fits in a single message for efficient and reliable transportation.

5.1.4. Piece Request Phase

Once the handshake phase is completed, the node starts to request pieces of data from those contactable peers which contain the piece/s. The node performs a piece selection process in order to choose the rarest piece from all the pieces contained in the list of peers. The rarest piece from the remaining pieces is acquired first in order to foster a uniform piece distribution and a quick piece dispersion over the network. This minimizes the risk of a data file not being completed, i.e. some of the pieces unavailable. A peer selection process is then launched to select the peer from which to acquire the piece of data such as to increase fairness and make the data distribution process more uniform and efficient.

5.2. Centralized vs. Decentralized

The TinyTorrents protocol has been designed to operate in both a partially centralised and a decentralised manner. In the centralised approach, a central node, i.e. the tracker, keeps control of the peers involved in the swarm of each torrent. On request from a peer wishing to join the swarm, the tracker node selects the list of peers which the requesting node should employ to fetch the data. In the centralised version, the benefits of having a central point for data distribution management are diminished by the reduced scalability of the protocol and the lack of information about the localization of peers. The decentralised version of the TinyTorrents protocol has been designed as a fault-tolerance solution which eliminates the unique central tracker and makes the system scalable. The decentralised design is based on the concept of every consumer node (peer involved in the data distribution process) acting as an opportunistic unstructured local point of coordination/information (i.e. partial tracker). The concept of “partial tracker” arises from the fact that the node only keeps track of a subset of the peers in the swarm within a nearby area/scope. The scope is not a fixed value and depends on the distribution of consumer nodes in the network, amongst other factors. A two-tier approach is employed to achieve a decentralised and scalable design. In the higher-tier, the TinyTorrents protocol coordinates the gradual distribution of data files over the whole network. At the lower-tier, the UMG routing protocol is in charge of reliably transporting data packets within the local scope of each node (a few hops away).

The decentralised version of the TinyTorrents protocol employs a set of unstructured discovery mechanisms for efficient location of partial trackers (consumers) capable of providing a list of peers from which to download the data file (see Section 5.5). Different peer selection strategies have been explored for the retrieval of a list of peers from tracker nodes and for the selection of peers from which data pieces are acquired (see Section 5.4). The mechanisms operate at the TinyTorrents layer and leverage UMG routing information and functionality.

5.3. Scalability of the Decentralized TinyTorrents System

The design of the decentralised version of the TinyTorrents protocol tackles some of the issues encountered in the centralised version with the use of “partial trackers”. The central tracker, in charge of maintaining the swarm of peers for all the torrents in the network, is replaced by consumer nodes acting as partial trackers only for their stored torrents. A consumer acts as a partial tracker for a given torrent by keeping track of a list of peers which do not necessarily need to comprise the whole swarm of the torrent. A partial tracker discovers and updates peers when i) the torrent data is being fetched, ii) peer list request messages are received, or iii) messages in the protocol are intercepted which contain the key of the torrent. Each consumer node is capable of acting as a partial tracker for nearby consumers. Thus peer lists are formed from a set of peers which are likely to be close to the requesting node. This behaviour can be seen in Figure 2 where consumer nodes send their “PeerListRequest” messages to nearby consumers acting as partial trackers.

In Figure 2, the cardinality in the consumer label (“C#”) sorts consumers according to the time when they start acquiring the peer list for the torrent. For instance, “C2” requests a peer list from the initial seeder “P” which only contains “C1” and itself as peers for the torrent. By the same token, “C8” selects “C5” as its partial tracker. “C5” contains in its swarm, at least, the peer list provided by “C4” which includes “C3”, “C2”, “C1” and “P’. This is due to the default size of the peer list being set to 4 and “C2” and “C1” being in the swarm of “P” when “C3” requests the peer list. In addition, “C5” acts as a partial tracker for “C7” which has also been included in its swarm of the torrent. For memory efficiency purposes, the maximum number of peers in the swarm of a partial tracker for a torrent is set by default to 10. When the swarm is full and new peers are discovered for the torrent, the current policy replaces the furthest peer (in hops) if the new peer is closer. This mechanism tends to populate the swarm of each partial tracker with closer peers. In this regard, a consumer needs to locate a nearby partial tracker to receive an efficient peer list in terms of proximity. Hence, the scalability of the system only depends on the distribution of consumers over the network, such that the highest distance between any two closest consumers is equal or less than the discovery scope of the routing protocol. For instance, in Figure 2, “C2” is within the routing scope of its partial tracker, i.e. “P”. However, “C10” is an isolated consumer due to the fact that the closer partial tracker is “C6” which is not within its routing scope; in other words, “C10” does not contain routing information about “C6” in its routing table. Therefore, the successful discovery of a nearby partial tracker is the key mechanism for the efficient operation of the decentralised version of the TinyTorrents protocol. In this regard, a set of mechanisms to discover partial trackers are presented in Section 5.5.

5.4. Peer Selection Strategies

In the TinyTorrents protocol appropriate selection of the peers from which to gather pieces of data impacts the efficiency of the traffic flow in terms of fairness, throughput, resource consumption and communications reliability. Peer selection strategies contribute to make the system scalable, to foster data dispersion, and to avoid the flash crowd problem at initial seeders.

5.4.1. Peer List Selection Strategies

In a partial tracker node, the selection of a peer list from the swarm of a torrent is mainly performed according to the chronological position at which a peer entered the swarm of the torrent for the tracker (position-based selection). The peer list selection strategy is a stochastic process where the distribution of probabilities amongst electable peers is based on their position in the swarm of the torrent. It is a weighting scheme where the probability of a peer “i” being selected for inclusion in the peer list “$P_{selPos}(i)$” can be calculated according to Equation 1,

$$P_{selPos}(i) = \frac{Position(i)^{\log(\alpha)}}{\sum_{x \in S} Position(x)^{\log(\alpha)}} \tag{1}$$

where “Position(i)” is the position at which the peer “i” entered the swarm of the torrent in the tracker. The denominator is computed as the sum of the weighted positions where “S” is the set of peers in the swarm except from those peers which could have been listed as previously failed in the peer list request message. Additionally, in order

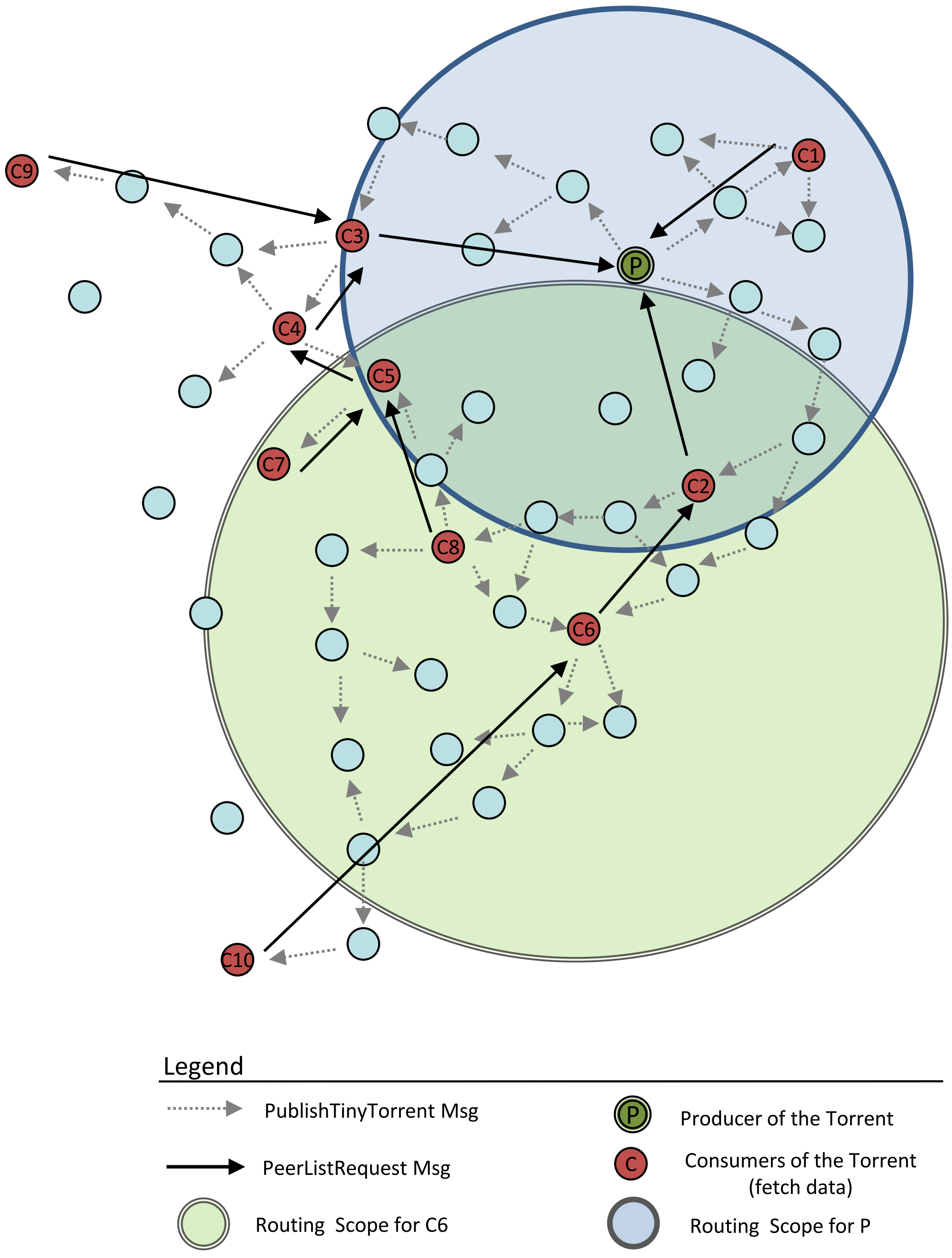

Figure 2: Operation of the Decentralised Version of the TinyTorrents Protocol

to increase the probability of selecting peers with higher positions, i.e. recently received peers, over peers which have been longer in the swarm, a coefficient α is utilised as a weighting parameter to control the growth of the position-based probability distribution. In the situation where a peer can not complete the data file with the first requested peer list and needs to request another peer list, the tracker can lower the α coefficient to the minimum value (α can take values from 1 to 255). A lower value of α increases the likelihood of peers with lower positions being selected under the assumption that, the lower the position in the swarm, the higher the likelihood for the peer to contain most of the pieces of data. However, when the peer list is requested for the first time for a given torrent, the value of α is set high in order to increase the probabilities of including recently joined peers in the peer list. While this is employed to achieve fairness in the data distribution process, including peers which recently joined the swarm in the peer list is of interest in the decentralised version since most of the torrent dissemination strategies would expand concentrically from the producer. This means that the likelihood of two nodes being in close proximity would increase as peers have a close value of position in the swarm.

5.4.2. Peer Selection Strategies

A set of different strategies are proposed for the peer selection process when a peer is in the *Piece Request Phase*, i.e. it has already received a peer list from the tracker, *Peer List Request Phase*, and has also performed the *Handshake Phase* with all, or some, of the peers in the list. In this phase, the peer has stored updated information on a set of peers from which to acquire pieces. For each contactable peer, the requesting peer contains the following information for the torrent: i) the piece bit vector and ii) the position in the swarm. Additionally, proximity to each of the contactable peers in terms of hops is known via the UMG routing protocol. While the position in the swarm and the piece bit vector of each peer are initially acquired in the *Handshake Phase*, the piece bit vector of a peer is updated with every message received in the *Piece Request Phase*. Thus, the peer selection process is executed when a new piece is to be retrieved as the current piece status of peers might have changed. This way, information about peers in the list is updated as the data fetching process progresses, which enhances the selection process in order to achieve fairness in the data distribution.

While the position in the swarm of a peer is a relevant factor to achieve fairness in a distributed manner and to alleviate the burden in initial seeders, the hop distance to each peer is the key factor to control the communications overhead and therefore the overall performance of the protocol. In single channel wireless sensor networks, a peer at a hop distance of 1 should always be chosen over 2 hops or more, to reduce the contention factor in the neighbourhood. However, when the peer selection process involves peers at a distance greater or equal to 2 hops, the unknown spatial distribution of the peers in their areas and the unknown contention conditions are insufficient information to assert that the closer peer in terms of hops becomes the most efficient choice. In this situation, the position in the swarm of the peer should be taken into account in the peer selection process.

The performance of a set of peer selection strategies under different configurations of the TinyTorrents protocol and network scenarios was studied. The highest degree of fairness in the data distribution process was obtained with strategies selecting peers with the most number of remaining pieces to be acquired. The strategies primarily seek to balance the flow of traffic towards those peers with a low number of pieces in order to reduce the load on seeders. However, strategies selecting proximate peers produced the best results in terms of communications, with a high degrees of fairness. The latter is due to the decentralized design of the system where partial trackers progresively distribute data amongst close consumers as the network scales. As a result, the Closest First Piece Remaining-based (CFP) Peer Selection strategy produced the most efficient results in terms of communications with high degree of data distribution fairness amongst peers. The CFP strategy firstly includes peers with the minimum number of hops from the list of contactable peers which have the piece to be requested. Then, the peer is selected by using the "$P_{selPos}(i)$" in Equation 1) where "Position(i)" is replaced by "PieceRemaining(i)". PieceRemaining(i) indicates the remaining number of pieces to be acquired by peer "i" according to its piece bit vector. The higher the number of pieces to be acquired, the higher the probabilities for the peer to be selected. This strategy achieves efficiency in terms of communication while at the same time serves to update peers in need of pieces of data on the availability of pieces at other peers, thereby reducing potential subsequent peer list requests to a partial tracker. In addition, this strategy balances the load of traffic within the near proximities due to the fact that the piece bit vector of nearby peers is also updated at the requesting node when the piece is received; this increases the chances of nearby peers being selected to provide the remaining pieces of data.

5.5. Unstructured Discovery of Partial Trackers

A set of combined mechanisms for the discovery of partial trackers are presented for the scalable and efficient operation of the TinyTorrents protocol in its decentralised version.

5.5.1. Torrent Dissemination Control

This discovery mechanism can be classified as push-pull, where initial seeders push the torrent message to the network in a way that can optimise the query/searching process of requesting consumers. The application layer controls the flooding process of the torrent by deciding at each receiving node if and when to forward the message. The first message received is forwarded after a delay, and thus a reception window is opened for the admission of further messages of the same torrent type. In this way, the reception of duplicate torrent messages is leveraged to update the consumer on the set of closest partial trackers as messages come from spatially separated consumers. When a node becomes a consumer of the torrent, the torrent message is broadcast with its address appointed as the partial tracker. Additionally, consumer nodes assign a smaller value of delay as compared to the rest of the nodes, which increments the likelihood of potential consumers receiving a higher number of instances of the message coming from disparate partial trackers. When the torrent is selected to be fetched or forwarded, the address of the closest partial tracker in the array is selected. However, the closest partial tracker can be a consumer which is due to start the fetching process at a later stage and thus it can not provide a peer list yet. In this case, the node contacts the array of potential partial trackers in a round-robin fashion until one of them provides a valid peer list. A delay is introduced in between requests to i) wait for these nodes to become partial trackers, and ii) avoid overloading the network.

5.5.2. "PeerListRequest" Message Interception

This mechanism enhances the previous mechanism by intercepting the "PeerListRequest" message en route to the previously discovered partial tracker. Every relay node in the route towards the partial tracker checks whether it can act as a tracker for the requested torrent key. In this case, if the node can provide a peer list for the torrent, then it becomes the new partial tracker.

5.5.3. Routing Discovery Scope Increment

This mechanism is employed when the appointed partial tracker cannot be contacted, either because the end-to-end routing connectivity fails, or because the node is out of the routing discovery scope (i.e. its address is not in the routing table). In this situation, the TinyTorrents protocol instructs the routing layer to increase its default discovery scope while searching for the partial tracker node. Thus, the default routing discovery scope is incremented by a factor of 3 hops to try to reach the node. When and if a partial tracker is discovered, the "PeerListRequest" message is sent towards the node.

5.5.4. UMG Service Discovery

When neither the appointed partial tracker nor the initial seeder can be reached and the *Routing Discovery Scope Increment* mechanism fails or is not employed, then the TinyTorrents protocol makes use of the Ubiquitous Mobile Gradient (UMG) routing protocol service advertisement and discovery functionality in the search for potential partial trackers. Two mechanisms are available:

Local Discovery: This approach exploits local information in the routing table to provide a potential target node in the peer list search process. Currently up to a maximum of three nodes from the routing table are selected to be contacted which best match the description of the torrent.

Network Discovery: When the *Local Discovery* mechanism can not find potential trackers within the local routing table, the UMG routing protocol leverages the gradient spreading process to discover nearby nodes with a similar service description. Nodes receiving the gradient packet spread their own gradients if their service descriptor matches the descriptor of the torrent in the packet within a certain degree of accuracy in the comparison of the descriptors; the accuracy is also transported in the packet and indicates the percentage value of similarity when comparing the bits of the two descriptors, i.e. Bloom filters (see Section 4). The scope of the gradient is incremented when the discovery is not successful up to a maximum number of hops (the default scope is 5 with an increment of 3). This approach operates as a descriptor-based multicast *Routing Discovery Scope Increment* mechanism.

6. Performance Evaluation

This section provides an evaluation of the decentralized version of the TinyTorrents protocol operating above the UMG routing protocol on top of the CSMA/CA MAC protocol. The system has been implemented using the TinyOS 2.1 operating system [22]. A set of scenarios were developed for the TinyOS simulator, TOSSIM, to test the performance of the system under different network conditions where the density, noise floor and distribution of consumer nodes in the network were varied. Results showed the effectiveness of peer selection strategies where the Closest First Piece Remaining-based (CFP) peer selection strategy produced the best combined results in terms of fairness and communications. Consequently, the CFP strategy has been employed in the subsequent evaluation of the performance of the system: i) in a real testbed comprised of 64 motes (see Section 6.3), ii) in terms of scalability in the simulator (see Section 6.4) and iii) in comparison with other dissemination protocols (see Section 6.5).

6.1. Metrics

The list of metrics employed in the performance evaluation of the TinyTorrents protocol operating on top of the UMG routing protocol are now presented and defined:

1. Total Packets Sent: The number of packets sent by a node at the MAC layer.
2. Total Packets Received: The number of packets received by a node at the MAC layer.
3. Network Total Packets Sent: The sum of the Total Packets Sent by all the nodes in the network.
4. Network Total Packets Received: The sum of the Total Packets Received by all the nodes in the network.
5. Torrents Completed: The ratio of the number of unique torrents for which the data has been successfully acquired to the number of unique torrents received at each consumer node.
6. Average Time Data File Completion (A-TDFC): The average time it takes for a torrent received at a node to acquire the whole data file from the moment the torrent is successfully selected from the queue of received torrents.
7. Piece Messages Sent: The total number of data messages (Piece Message) sent by each consumer/producer node. This metric quantifies the involvement of a consumer/producer node in the data distribution process of a torrent, or set of torrents, as compared to the rest of the consumers in the swarm.
8. Jain's Fairness Index (JFI): The Raj Jain Fairness Index [80] is a metric employed to assess the degree of fairness in the resource utilisation within a set of nodes in the network. The JFI is computed by equation 2,

$$JFI(x_1, x_2, ..., x_n) = \frac{(\sum_{i=1}^{n} x_i)^2}{n \sum_{i=1}^{n} x_i^2} \tag{2}$$

   where "n" is the number of nodes, and "x(i)" the resource value being studied. In our experiments, "x(i)" corresponds to the total number of packets sent or received. The interpretation of the JFI result ranges from: i) $\frac{1}{n}$ (worst case scenario) to ii) 1 (best case scenario) where all nodes are allocated the same resource.

6.2. Methodology

An application has been developed which controls the distribution process of the TinyTorrents framework. A producer node indicates the data file which is to be distributed. Currently, the system is capable of distributing files of up to 255 bytes; this limit has been established considering the available RAM memory of the motes in the real testbed (i.e. 10 KB for the TelosB [81]). A data file of 255 bytes is subsequently divided by the TinyTorrents protocol into 16 pieces of data; by default the size of each piece of data is 16 bytes which has been shown to achieve an efficient balance between reliability in the transmission of the message and amount of data being transported. When a producer node starts the distribution process, the torrent file is disseminated through the network. In this phase, the application layer decides upon reception of a torrent message: i) whether the node is acting as a consumer of the data file and ii) whether the torrent file is to be disseminated (broadcast). The application layer also decides i) when the fetching process of the data needs to be started and, ii) when the torrent file is to be broadcast. A set of network topologies with regular and irregular layouts have been employed which scale from 64 to 400 nodes.

6.2.1. *Torrent File Dissemination Strategies*

The strategy for disseminating the torrent file through the nodes in the network is a key mechanism in the discovery of partial trackers in the decentralised version of the TinyTorrents protocol (see “Torrent Dissemination Control” in Section 5.5). The producer broadcasts the torrent message three times while the rest of the nodes broadcast it twice. In between broadcasts a delay of 300 milliseconds is introduced. In addition, a randomized time up to 200 ms is added to minimize the likelihood of collisions. For evaluation purposes the delay to forward the torrent message has been set to 1 second in consumer nodes while non-consumer nodes delay the forwarding for 3 seconds; this configuration has been selected as it produces a progressive dissemination of the torrent messages through the network containing the address of closer partial trackers.

6.2.2. *Consumer Distribution Strategies*

The number and the ditribution of consumers in the network is also a key factor which impacts the data distribution process. Having a high density of consumers will not exploit the benefits of the TinyTorrents selective data dissemination mechanism, but rather would suggest the use of epidemic mechanisms for data dissemination. On the other hand, if consumers are at a distance greater than the routing discovery scope, the TinyTorrents system launches its unstructured discovery mechanisms (see Section 5.5). Expanding the routing discovery scope increases the communications cost and, at high hop distances, the packet delivery ratio starts to drop. Thus, the strategy for selecting which nodes become consumers of a given torrent are paramount for the TinyTorrents system to be fully scalable. A balance needs to be achieved which takes into account the default routing discovery scope and the scope increment value employed by the unstructured discovery mechanisms to expand the searching scope.

Two types of distribution strategies have been created with different density of consumers and inter-consumer distance: i) Low Consumer Density and ii) High Consumer Density. For the 64 nodes topology, a low consumer distribution of 8 nodes have been created with a distance of 3 to 4 hops amongst them, namely 64_8 (see Figure 3). Distributions with high consumer density follow the “3ID” strategy in which nodes become consumers when their identifier is modulo 3 (see Figure 4); in the topology, nodes are numbered in ascending order from left to right and bottom to top.

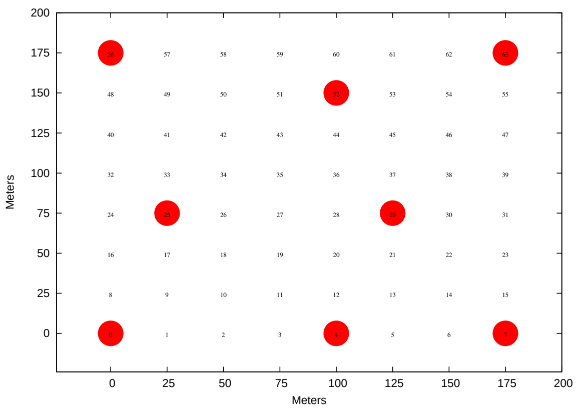

Figure 3: 64_8 Consumer Distribution Strategy. 64 Nodes SG Topology

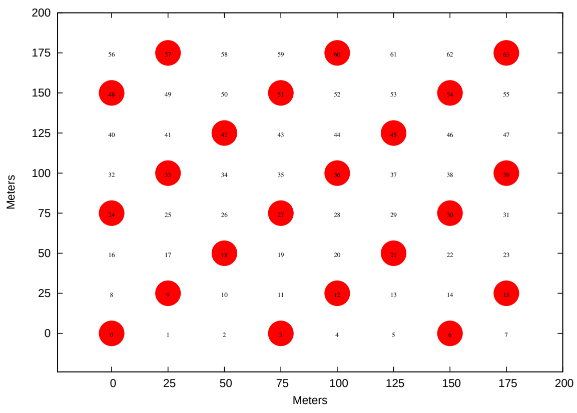

Figure 4: 64_3ID Consumer Distribution Strategy. 64 Nodes SG Topology

Two different topologies of motes were formed: i) the Square Grid (SG) layout, and ii) the Irregular Grid (IG) layout. The Square Grid topology follows a square grid layout where motes in the topology have been placed at 25 meters spacings vertically and horizontally (see Figures 3 and 4 for the 64 nodes SG topology). This topology has been employed in most of the experiments and particularly in the simulator, where the nodes’ transmission range has been set to 37 meters. On the other hand, the protocol has been evaluated in a real testbed operating on a Irregular Grid (IG) topology where nodes have been shifted randomly from the initial Square Grid layout (see Figure 9). Note that by default the UMG routing protocol is configured with a routing scope of 5 hops.

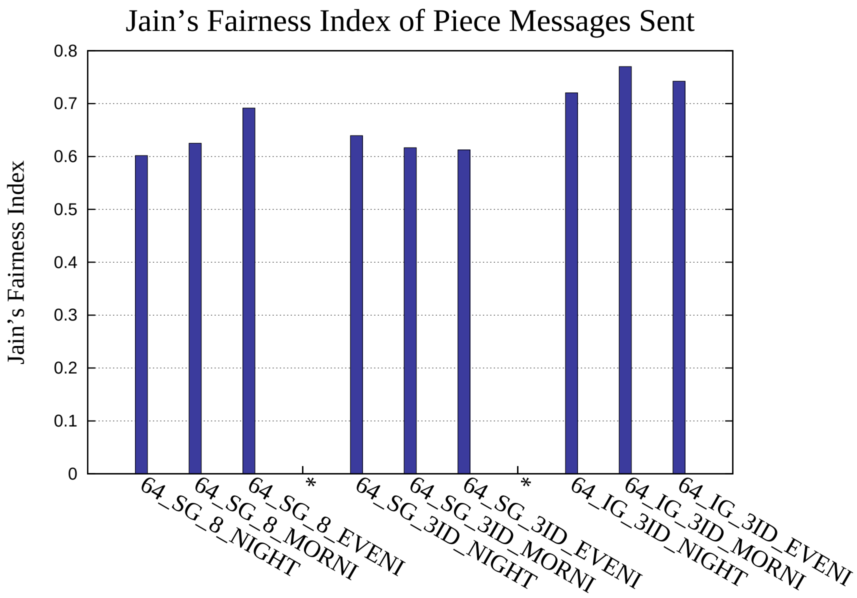

Figure 5: Jain's Fairness Index of Piece Messages Sent

6.3. Real Testbed Performance Evaluation

This section explores the performance of the decentralised version of the TinyTorrents protocol in a testbed of 64 TelosB [81] wireless sensor devices deployed in a home environment following the layout of the Square Grid (SG) and the Irregular Grid (IG) topologies and a distance between vertical and horizontal adjacent nodes of 30cm. In order to achieve multi-hop communication in the test environment, the lowest transmission power of the transceiver was set. According to the CC2420 chip specifications [82], the RF output power register was set to a minimum nominal value of 3, delivering -25 dBm. With this configuration nodes could communicate with 1-hop neighbours in all directions in the SG topology.

Three producer nodes have been appointed to periodically publish data files. Node 1, 38 and 49 have been selected as the three producers due to their distant location from each other in the network. Each producer publishes 20 torrents every 300 seconds. Producers are randomly scheduled to start publishing the next torrent within the first 20 seconds once the 300 seconds interval is elapsed. This generates simultaneous data file distribution processes for each of the 20 torrent intervals. Two deterministic consumer distribution strategies have been employed, the 64_8 and the 64_3ID, which enable comparison under different consumer densities, network traffic burden and inter-consumer hop distances. The size of the queues which store torrent information and data files have been reduced to fit the available RAM memory in the TelosB architecture, i.e. 10KB. A maximum of 4 data files are stored at each consumer/producer. Each experiment took, from the time the producer motes were started until the acquisition of the last torrent, an average of 6300 seconds (1h 47'). Three repetitions of each experiment have been carried out at different times of the day - morning (MORNI), evening (EVENI), overnight (NIGHT) - to account for different scenarios of background radio activity. The radio activity in the IEEE802.11(b,g,n) 2.4GHz frequencies increased in the evening as compared to morning and night time periods. This variation was also observed in the IEEE802.11 channel 13 (2472MHz) which operates at a closer frequency to that of the TelosB sensor devices, configured to communicate in the IEEE802.15.4 channel 26 (2480MHz). Also, it has to be noted that the default routing discovery scope has been set to 5 hops.

Results are computed for the 9 experiments performed in the testbed. Figure 5 shows the Jain's Fairness Index (JFI) of the number of Piece Messages Sent. The Jain's Fairness Index (JFI) in equation 2 has been employed to calculate the data traffic load of the set of nodes in the swarm of a torrent(s) from a given producer(s). This has been achieved by computing the JFI of the number of Piece Messages Sent by each of the consumers, including the producer, in each scenario. The index provides the degree of fairness in the data piece distribution process; the metric does not take into account the routing traffic at each node. The JFI worst case scenario is computed as $\frac{1}{n}$, where n

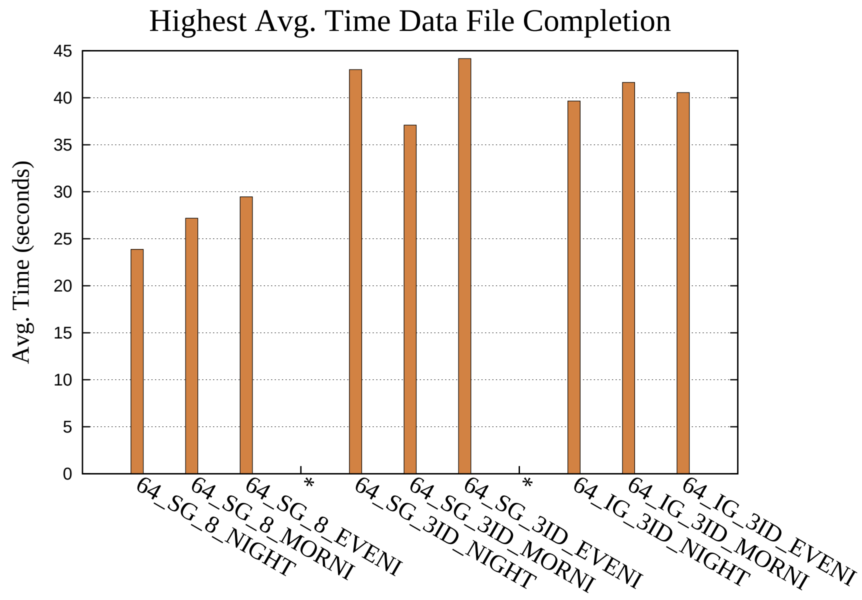

Figure 6: Highest Avg. Time Data File Completion

is the number of nodes in the swarm of the torrent capable of sending pieces of data (i.e. the number of consumers plus the producer). According to the number of consumers in each of the consumer distribution strategies employed, the JFI best case - worst case interval is: i) 64_8 [0.11-1] and 64_3ID [0.043-1]. However, values close to 1 are not expected mainly due to the fact that consumer nodes at the edges of the network, or at the end of the range of the data distribution process, might not even be required to provide pieces of data. Results show that, for all scenarios, JFI values above 0.6 are obtained which indicates a good degree of fairness in the distribution process between consumers, in line with results obtained in the simulator. Experiments with the Irregular Grid (IG) topology produce higher degree of fairness (JFI of Piece Messages Sent), which indicates the effect that the distribution of consumers in the network has for the fair distribution of pieces of data. No major difference is produced in the JFI of Piece Messages Sent at different times of the day. Furthermore, Figure 6 shows the highest average time achieved in the completion of the data file. As expected, the higher the number of consumers, the higher the average to complete the data file; this is due to the fact that torrents received from other producers are waiting in the queue until the current one is processed (i.e. data file acquired). In this figure, a difference of 5 seconds is not a significant variation to draw conclusions on the impact of the noise at different times of the day, mainly due to the fact that producers publish torrents at random times within the first 20 seconds of each sequential torrent.

The testbed experiments have proved the efficacy and reliability of the TinyTorrents decentralised protocol operating in a network of 64 TelosB devices in a home environment as the system (remarkably) achieved a 100% Torrents Completed ratio for all the experiments.

Additionally, Figures 7, 8 and 9 show the distribution of network traffic and average time to complete the data file for the experiment with the 64_3ID consumer distribution in the Irregular Grid topology performed in the evening (EVENI).

The Irregular topology makes the distribution of consumers more suitable to achieve fairness than the consumer layout achieved in the Square Grid topology which is indicated by the JFI values (0.63 vs. 0.74) in Figure 5. This confirms the impact the consumer distribution has in the efficiency of the dissemination process. In Figure 7 node 18 is a hot spot in the distribution of pieces of data. This is expected due to its key position, acting as a central gateway node between consumers in the north and south of the network. In terms of total packets received, both topologies depict the same expected pattern where central nodes receive most of the packets; central nodes are surrounded by a higher number of neighbours and are required to route more packets. In Figure 8, the highest number of total packets received correspond to central nodes with a high number of close consumers, but also to those nodes in the gateway

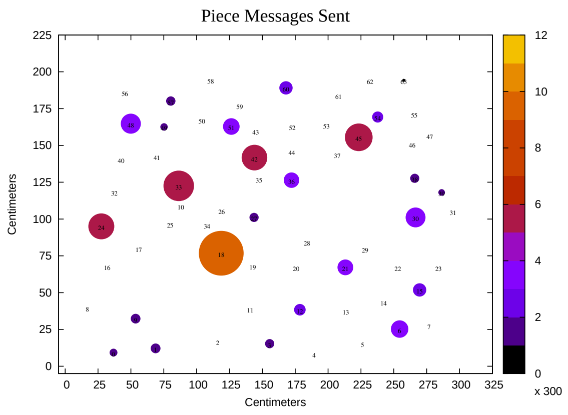

Figure 7: Piece Messages Sent - 64_3ID, IG - JFI 0.74

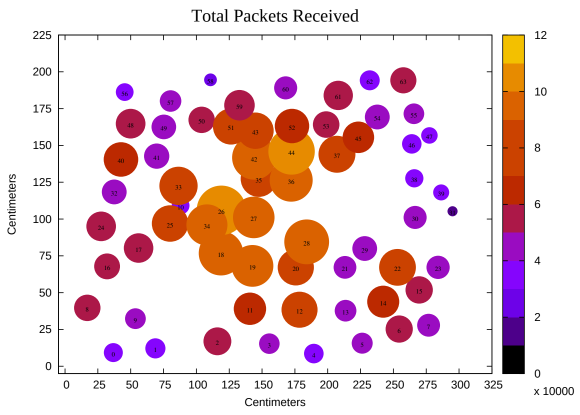

Figure 8: Total Packets Received - 64_3ID, IG - JFI 0.91

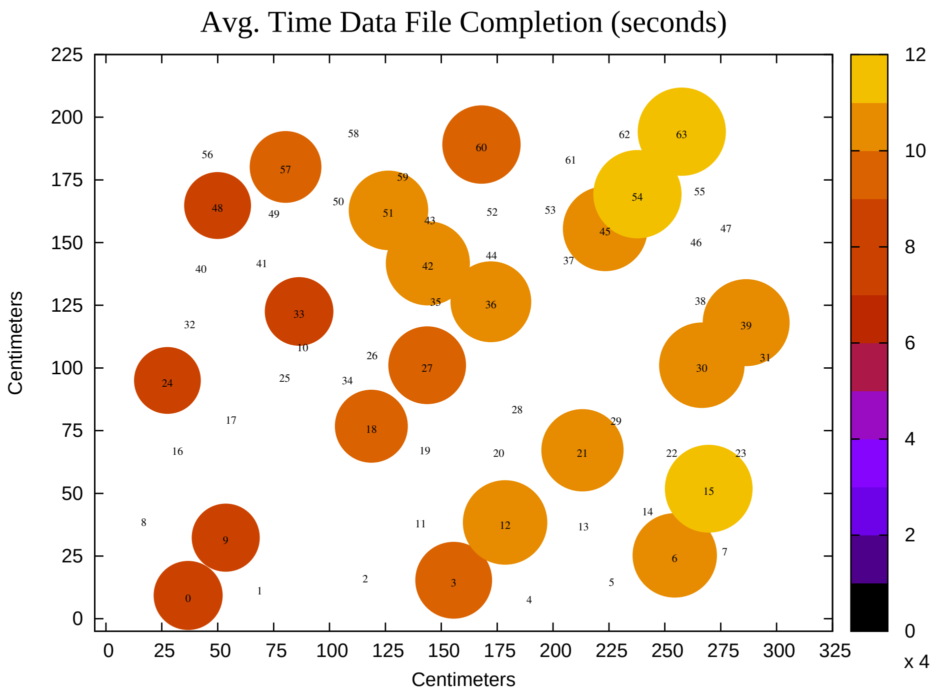

Figure 9: Avg. Time Data File Completion - 64_3ID, IG

path where node 18 is located. Some nodes at the edges of the Irregular Grid layout exhibit a high number of packets received. These nodes are employed as key routers to reach certain areas; for instance node 8. The JFI of the Total Packets Received is high (0.91) which is expected for this type of scenario where a high density of consumers and three producers placed at distant points generate traffic in most of the areas of the network. The traffic load generated by the high number of consumers has an impact on the average time to acquire a data file (see Figure 9 which takes on average between 27 and 44 seconds. The lowest average times to acquire the data file in the Irregular Grid topology correspond to nodes placed at the west side of the network. This is explained by noting that two of the producers are placed in the west side of the network (1 and 49) and the other producer (node 38) is placed at the east side, together with the fact that central nodes are farther from node 38 in terms of routing.

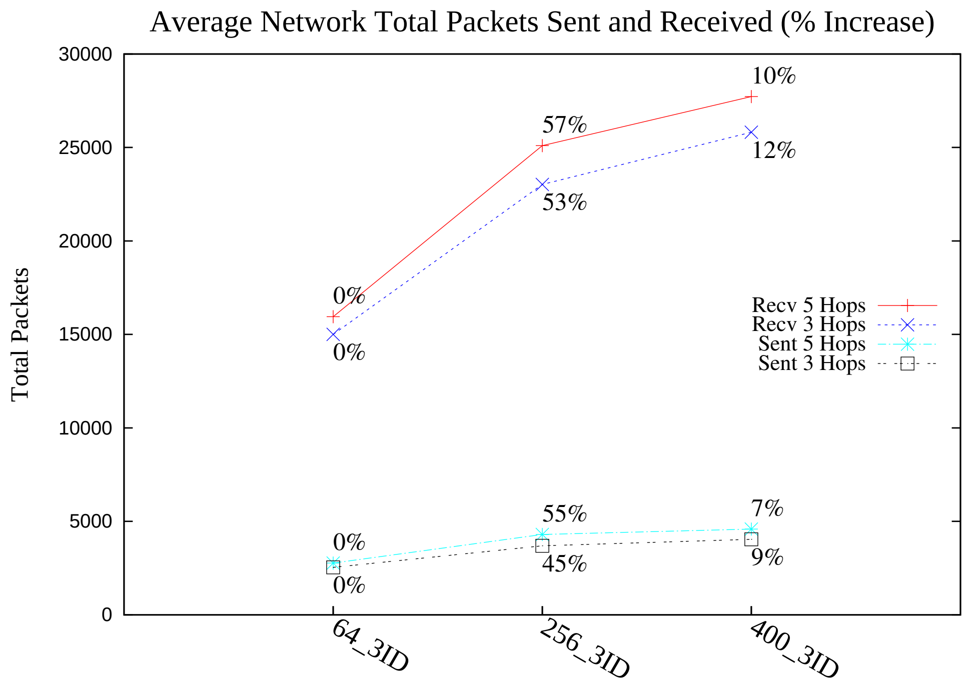

Figure 10: On Scalability - Average Network Total Packets Sent and Received (% Increase)- Routing Scope of 3 and 5 Hops - Avg. of 5 repetitions

6.4. Scalability of the TinyTorrents System

This section evaluates the performance of TinyTorrents when the network scales using the TOSSIM simulator. The scalability factor of the TinyTorrents protocol depends on the distribution of consumers in the network and the maximum routing scope. A consumer at a larger distance than the maximum routing scope from the closest consumer will be disconnected from the network of consumers and thus will not be able to fetch the data file. Moreover, a consumer will not be able to fetch the data file until at least one of the consumers within its routing scope acquires the data file. Thus, when scaling the data file distribution process, the distribution and number of consumers impacts the Torrents Completed ratio, as well as the distribution of Total Packets Sent and Received by each node in the network.

For the purpose of comparing the scalability performance of the decentralised version of the TinyTorrents protocol operating with the CFP peer selection policy, the square grid (SG) topology has been scaled from 64 to 256 and up to 400 nodes. The node transmission range has been set to 37 meters and a moderate-to-high noise floor (Casino noise trace in TOSSIM) has been utilized. One producer, Node 1, generates 20 torrents every 100 seconds. The 3ID consumer distribution strategy has been selected as it produces a similar consumer distribution structure and enables comparison when the network scales. Each scenario has been configured with two routing scopes, 3 and 5 hops, to evaluate its impact when the network scales.

In order to compare the cost in packets of scaling both the network and the consumer distribution, the Average Network Total Packets Sent and Received metrics, calculated as an average of 5 repetitions, have been plotted in Figure 10. The figure also shows the percentage increase with respect to the previous value. Due to the high density of consumers in the 3ID strategy, the average network total packets sent and received increases with scalability. However, a higher percentage increase is obtained when scaling from the 64_3ID to the 256_3ID scenario - 57% (Recv), 55% (Sent) -, as compared to scaling from the 256_3ID to the 400_3ID scenario - 10% (Recv), 7% (Sent). This is due to the lower proportion of nodes at the edges of the square topology as the network scales. Nodes at the edges do not get as much participation in the network as central nodes and thus their proportion in the network impacts the average network total packets sent and received. In addition, a higher number of consumers fetching the data file within the routing scope increases the overall network total packets received and sent. In this regard, lowering the routing scope

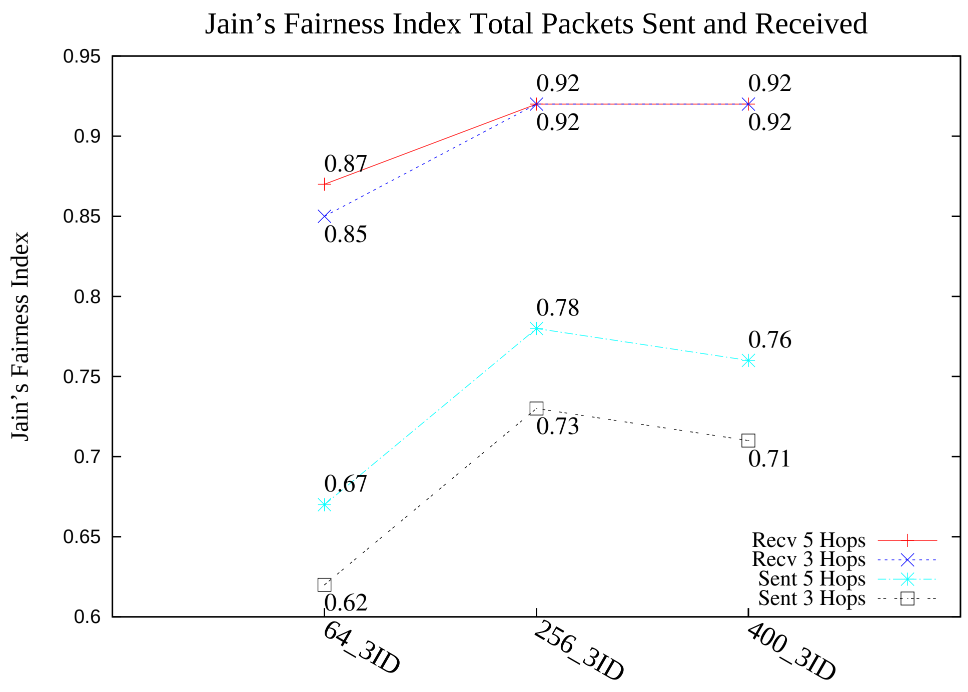

Figure 11: On Scalability - Jain's Fairness Index of the Total Packets Sent and Received - Routing Scope of 3 and 5 Hops - Avg. of 5 repetitions

from 5 to 3 hops, lowers the average network total packets sent and received (see Figure 10) by a percentage decrease in the range of 6-9% (Recv) and 8-16% (Sent).

Figure 11 shows the Jain's Fairness Index of the Total Packets Sent and Received calculated as an average of 5 repetitions. The JFI has been computed for all the nodes in the network in terms of total packets sent and received to assess the degree of fairness in the overall network communication process for each scenario. However, it has to be noted that the JFI of the Total Packets Sent and Received is impacted by the number and distribution of consumers in the network. Thus, the metric serves to evaluate the scalability of a network when it grows with a scalable consumer distribution, such as the 3ID. As expected, Figure 11 shows higher index of fairness for the total packets received than for the total packets sent. This is due to the broadcast nature of the wireless medium. In addition, higher values of JFI are obtained for the 256 and 400 nodes topologies as compared to the 64 nodes topologies. This is mainly a consequence of the lower proportion of nodes at the edges and the higher number of consumers at the centre of the network in the 256 and 400 nodes topologies. The high values of JFI in the total packets received (Recv) are due to the high density of consumers in the network, but they also confirm that the number of packets received is balanced. The JFI of the total packets sent (Sent) has lower values due to the act that the 3ID consumer distribution strategy places a great proportion of consumers at 1 hop distance and consequently reduces the participation of non-consumer router nodes and hence their packet sending ratio. By the same token, the higher the routing scope, the higher the likelihood of non-consumer nodes acting as routers and thus contributing to increase the Jain's fairness index of the Total Packets Sent (see "Sent 5 Hops" vs. "Sent 3 Hops" in Figure 11).

Results indicate that the decentralised version of the TinyTorrents protocol scales at a low cost in terms of total packets received and sent, which depends on the routing scope. It also exhibits high values of fairness in the communications process in the network, regardless of the scale of the network.

6.5. Performance Comparison of Dissemination Protocols

This section analyses the performance of TinyTorrents when compared to two of the most popular dissemination protocols for Wireless Sensor Networks, i.e. DIP [25] and DHV [26] (see Section 2). DIP and DHV have been designed with the goal of reprogramming the network in an epidemic fashion such that consistency of data is achieved amongst all the nodes in the network. This mechanism of dissemination does not follow the selective data dissemination approach that underpins the design of TinyTorrents. While DIP and DHV are good solutions for network reprogramming purposes, TinyTorrents provides for the distribution of data amongst a subset of consumer nodes in the network which express interest in the data, and thus the traffic load in the network depends on the consumer distribution. The higher the number of consumer nodes in the network, the more efficient the use of an epidemic dissemination protocol. However, consider a scenario where consumers and producers are placed within a defined area in the network for the purposes of performing some sort of sensing-actuating activity, for instance at the edge of the network. Nodes in other areas would not need to receive data from these producers and consumers, and the data transfer should only occur within this area and, most specifically, amongst the overlay of consumers and producers. TinyTorrents provides this type of selective data dissemination where router nodes do not necessarily need to receive all the pieces of a file of data, thus also increasing the security of the process. These advantages need to be taken into consideration when comparing TinyTorrents with epidemic dissemination protocols which have been shown to be very efficient in terms of communication.

For the purpose of providing a fair comparison in the TOSSIM simulator, an application running on top of DIP and DHV has been developed to disseminate the same amount of data as the TinyTorrents application. One producer, Node 1, has been selected to disseminate 20 files of data, each of size 255 bytes. The Piece message which contains the data in the TinyTorrents protocol, has been encapsulated in the payload of these protocols. The Piece message has been configured to contain 16 bytes of data and has been fitted in a packet by increasing the payload to that used for TinyTorrents packets. DIP and DHV operate by updating versions of data items where the most up-to-date version is disseminated with the goal of maintaining consistency in the network. Data packets corresponding to an old-version are not disseminated when a new version, i.e. an update, for the data item is, or has been, received at the node. Taking this into consideration, the application assigns each of the 16 pieces of a file a different key, such that they are described as different data items. In this way, different versions of each piece are injected into the network when a new file is disseminated. The same delay employed in TinyTorrents for the dissemination of files by the producer is also in place which guarantees that no pieces of data are being transferred when a new version, i.e. a new file, is starting to be disseminated. This mechanism enables comparison with TinyTorrents where each piece of data is distributed as a different data item. Note that one of the drawbacks of DHV and DIP is the requirement for defining the type of data items to be disseminated at compilation time, such that all the nodes know beforehand what type of data items are to be received.

Additionally, DIP and DHV employ the Trickle algorithm [27] to regulate the periodicity of broadcasting packets. Trickle exponentially increases the packet broadcast interval to reduce communications according to the activity in the neighbourhood, while decreasing the interval towards a minimum value, T_l, when updates need to occur. Setting T_l to a low value will increase the communications at the benefit of reducing the latency in the dissemination process. T_l has been set to 30 ms; a lower value produces similar results in terms of latency and employs more packets. In this way, both DHV and DIP have been configured to achieve a very high performance in terms of latency with a low overhead in communications. To further improve latency on DIP and DHV, pieces are sequentially published by the producer with a small delay of 50 milliseconds. On the other hand, the TinyTorrents protocol employs a set of control messages, such as handshakes and peerlist request messages, to regulate the dissemination process. This enables control of the distribution of the traffic but also increases the latency and packet overhead in the dissemination of data.

The scenario selected for the comparison comprises 64 nodes in a square grid (SG) layout, a node transmission range of 37 meters and a moderate to high noise floor (Casino noise trace in TOSSIM). To show the benefits of the selective data dissemination process of the decentralised version of the TinyTorrents operating with the CFP peer selection strategy, two consumer distributions have been employed: i) the 64_CONS8 distribution which employs 8 consumers placed across the network, and ii) the 64_CONS8E distribution, a new distribution created for this purpose which places 8 consumers along the bottom edge of the network (nodes 2, 4, 7, 10, 12, 14, 15, 19) (see Figure 13). Consequently, 4 protocols configurations have been evaluated: i) TinyTorrents with the CONS_8 distribution, ii) TinyTorrents with the CONS_8E distribution, iii) DIP with $T_l = 30$, and iv) DHV with $T_l = 30$. Note that, as DHV and DIP are epidemic dissemination protocols, they deliver data to all nodes in the network and thus all nodes are consumers.

Latency in the dissemination process has been studied with the Average Time Data File Completion (A-TDFC) in Figures 12 and 13 for TinyTorrents, in Figure 14 for DIP and in Figure 15 for DHV. For DIP and DHV, the A-TDFC

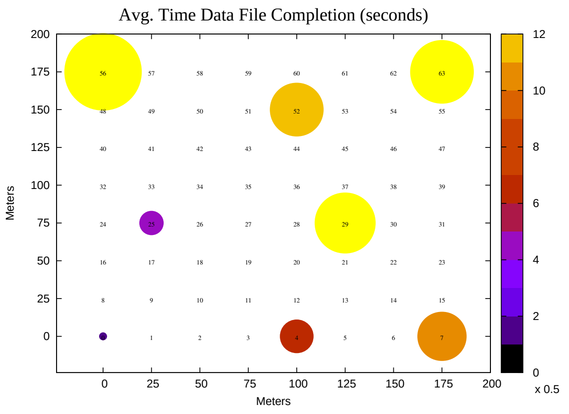

Figure 12: TT Decentral - Avg. Time File Completion (seconds) - CONS_8

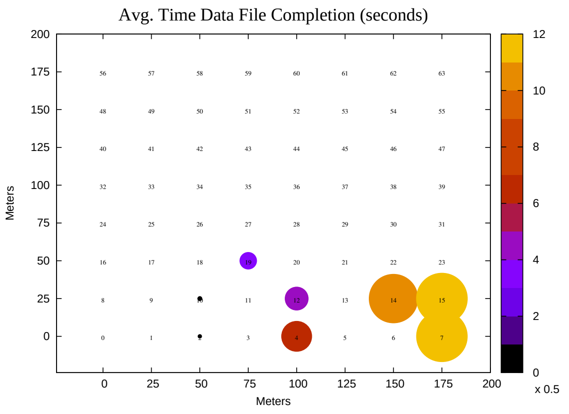

Figure 13: TT Decentral - Avg. Time File Completion (seconds) - CONS_8E

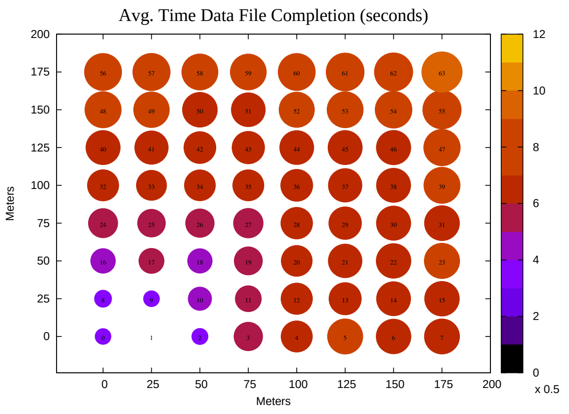

Figure 14: DIP - Avg. Time File Completion (seconds) - T$_l$ = 30

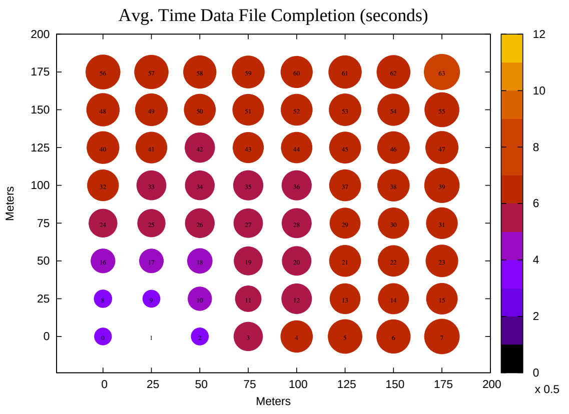

Figure 15: DHV - Avg. Time File Completion (seconds) - T$_l$ = 30

for the furthest nodes from node 1, i.e. the producer, is of the order of 3 to 4 seconds, where DHV shows faster file completion times than DIP. TinyTorrents also shows a faster data file completion time for closer nodes such as 0 and even 4 in Figure 12, and nodes 2, 10, 12 and 19 in Figure 13. However, as the file is disseminated through the network, the data file completion time increases, producing higher values than with DIP and DHV. This is mainly due to the delay introduced by the consumers when waiting for other consumers within their scope to retrieve the file. For example, node 7 takes double the time to acquire the file than it takes with DIP and DHV. This result demonstrates that epidemic algorithms can offer faster solutions for delivering data, while employing less control packets, at the cost of disseminating the file throughout the whole network.

Figures 16 and 17 show the Total Packets Sent and Received respectively for each of the 64 nodes in the network for the 4 protocols configurations. DIP and DHV show evenly distributed network traffic, both for Total Packets Sent and Received. This is due to the Trickle algorithm which regulates packet transmissions according to the neighbourhood activity while leveraging broadcast packets for the update of data items. DHV requires less packets than DIP, while showing similar behaviour. By comparison, TinyTorrents employs higher number of Total Packets Sent, mainly by those nodes which fully participate in the data dissemination process, both router and consumer nodes. However, TinyTorrents sends fewer packets than DIP and DHV from those nodes placed in areas where there is a lower concentration of consumers. This effect is more clearly visible with the CONS_8E consumer distribution, where nodes are placed at the edge, with the middle top area of the network (node 30 to 63) not receiving data and only receiving torrent messages. This shows the efficiency of TinyTorrents as a selective data dissemination protocol and the unsuitability of epidemic dissemination in this case.

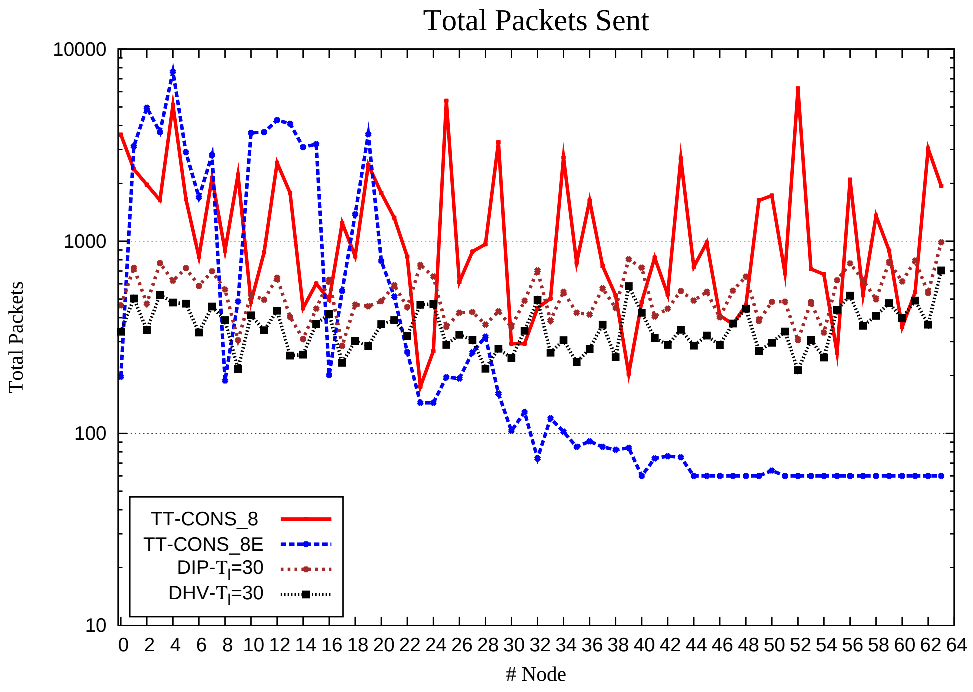

Figure 16: Dissemination Protocols Comparison: TT, DHV and DIP - Total Packets Sent - 64 Nodes, 1 Producer , SG, 37m, Casino, TT Decentral, CFP

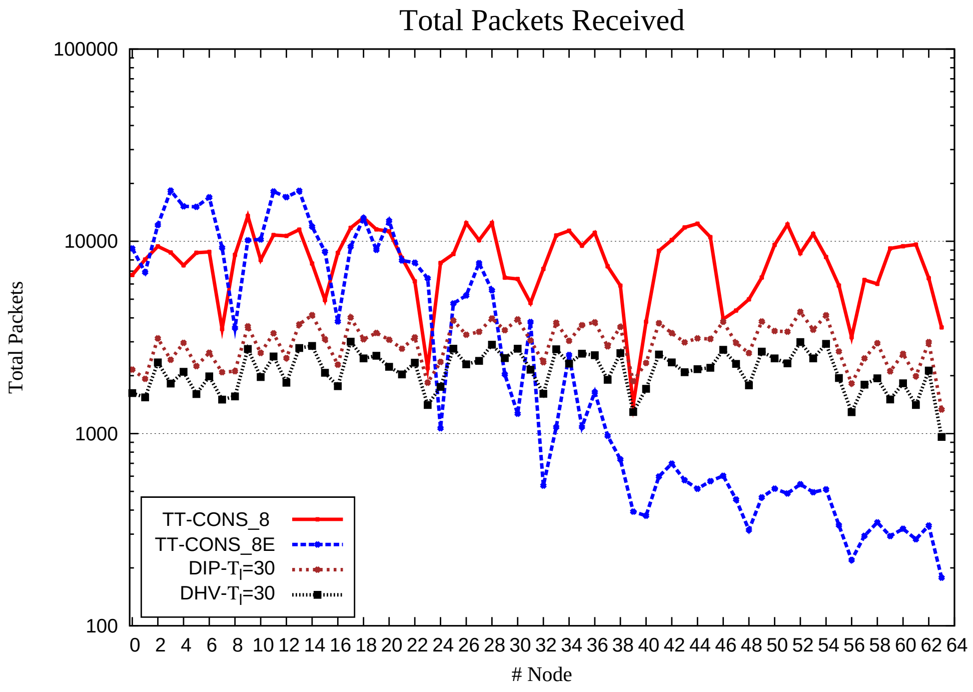

Figure 17: Dissemination Protocols Comparison: TT, DHV and DIP - Total Packets Received - 64 Nodes, 1 Producer , SG, 37m, Casino, TT Decentral, CFP

7. Conclusion

A novel communications architecture which provides scalable decentralised selective data dissemination in wireless sensor networks has been presented. The TinyTorrents architecture validates a solution to the following research problem: how to reliably disseminate data to a sparse subset of interested nodes in an unstructured, scalable wireless sensor network whilst distributing the traffic load. The proposed solution addresses the problem by employing peer-to-peer content distribution concepts where data is distributed to a subset of nodes in the network in a collaborative way. A decentralised communications system composed of a data distribution layer operating above a routing protocol has been presented to tackle such problem. The rationale behind the approach is based on a high layer coordinating data acquisition and distribution behaviour amongst the overlay of proximate consumer/producer nodes, whilst employing a reliable gradient-based routing protocol for nearby multi-hop data transportation. A set of unstructured discovery mechanisms which leverage routing information are employed to locate other consumers and peer selection strategies efficiently select closer consumers to distribute the traffic load. In addition, the TinyTorrents solution transparently integrates, in a decentralised fashion, wireless sensor networks with the global BitTorrent peer-to-peer network. In doing so it seemlessly interconnects deeply embedded, ubiquitous devices and networks with any contactable Internet agent.

Simulated and practical evaluation has shown that the system reliably, scalably and efficiently disseminates data within the wireless sensor network and beyond. The architecture has proved to be an effective and scalable communication substrate for the development of cooperative applications in sensor networks. These type of applications can be characterised by consumer and producer nodes belonging to multiple overlays of interest, akin to social networks today, where data needs to be distributed to the members of each overlay. Complex behaviours could arise from networks of sensor and actuators performing basic tasks on data which is either acquired from sensing or received from consumers/producers in an overlay of interest.

8. Acknowledgements

This publication has emanated from research conducted with the financial support of Science Foundation Ireland.

9. References

[1] W. Weber, J. Rabaey, E. Aarts, Ambient Intelligence, Springer Verlag, 2005.

[2] C. McGoldrick, M. Clear, R. Simon Carbajo, K. Fritsche, M. Huggard, TinyTorrents-Integrating Peer-to-Peer and Wireless Sensor Networks, Sixth International Conference on Wireless On-Demand Network Systems and Services, WONS 2009. (2009) 119–126.

[3] R. S. Carbajo, E. S. Carbajo, B. Basu, C. Mc Goldrick, Routing in wireless sensor networks for wind turbine monitoring, Pervasive and Mobile Computing 39 (2017) 1–35.

[4] C. Intanagonwiwat, R. Govindan, D. Estrin, Directed diffusion: a scalable and robust communication paradigm for sensor networks, Proceedings of the 6th annual international conference on Mobile computing and networking (2000) 56–67.

[5] W. Heinzelman, J. Kulik, H. Balakrishnan, Adaptive protocols for information dissemination in wireless sensor networks, Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking (1999) 174–185.

[6] J. Kulik, W. Heinzelman, H. Balakrishnan, Negotiation-based protocols for disseminating information in wireless sensor networks, Wireless Networks 8 (2/3) (2002) 169–185.

[7] D. Estrin, R. Govindan, J. Heidemann, S. Kumar, Next century challenges: Scalable coordination in sensor networks, Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking (1999) 263–270.

[8] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, F. Silva, Directed diffusion for wireless sensor networking, IEEE/ACM Transactions on Networking 11 (1) (2003) 2–16.

[9] H. Luo, F. Ye, J. Cheng, S. Lu, L. Zhang, TTDD: two-tier data dissemination in large-scale wireless sensor networks, Wireless Networks 11 (1) (2005) 161–175.

[10] D. Coffin, D. Van Hook, S. McGarry, S. Kolek, Declarative ad-hoc sensor networking, Proceedings of International Symposium on Optical Science and Technology 4 (2000) 109–120.

[11] F. Ye, G. Zhong, S. Lu, L. Zhang, Gradient broadcast: A robust data delivery protocol for large scale sensor networks, Wireless Networks 11 (3) (2005) 285–298.

[12] M.-G. Lee, S. Lee, Data dissemination for wireless sensor networks, 10th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC'07) (2007) 172–180.

[13] Z. Can, M. Demirbas, A survey on in-network querying and tracking services for wireless sensor networks, Ad Hoc Networks 11 (1) (2013) 596–610.

[14] C. Wan, A. Campbell, L. Krishnamurthy, Pump-slowly, fetch-quickly (PSFQ): a reliable transport protocol for sensor networks, IEEE Journal on Selected Areas in Communications 23 (4) (2005) 862–872.

[15] S. Park, R. Vedantham, R. Sivakumar, I. Akyildiz, A scalable approach for reliable downstream data delivery in wireless sensor networks, Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing (2004) 78–89.

[16] F. Stann, J. Heidemann, RMST: Reliable data transport in sensor networks, IEEE International Workshop on Sensor Network Protocols and Applications (2003) 102–112.

[17] T. Stathopoulos, J. Heidemann, D. Estrin, A Remote Code Update Mechanism for Wireless Sensor Networks, Tech. rep., CENS-TR-30, University of California, Los Angeles, Center for Embedded Networked Computing (2003).

[18] S. Kulkarni, L. Wang, MNP: Multihop network reprogramming service for sensor networks, Proceedings of the 25th IEEE International Conference on Distributed Computing Systems, ICDCS 2005. (2005) 7–16.

[19] M. Doddavenkatappa, M. C. Chan, B. Leong, et al., Splash: Fast data dissemination with constructive interference in wireless sensor networks., NSDI (2013) 269–282.

[20] Z. Zhao, W. Dong, J. Bu, Y. Gu, C. Chen, Link-correlation-aware data dissemination in wireless sensor networks, IEEE Transactions on Industrial Electronics 62 (9) (2015) 5747–5757.

[21] M. Mousavi, H. Al-Shatri, M. Wichtlhuber, D. Hausheer, A. Klein, Energy-efficient data dissemination in ad hoc networks: Mechanism design with potential game, 2015 International Symposium on Wireless Communication Systems (ISWCS) (2015) 616–620.

[22] TinyOS Alliance, TinyOS version 2.x (T2), `http://www.tinyos.net`.

[23] J. Hui, D. Culler, The dynamic behavior of a data dissemination protocol for network programming at scale, Proceedings of the 2nd international conference on Embedded networked sensor systems (2004) 81–94.

[24] C. Liang, R. Musăloiu-E, A. Terzis, Typhoon: A reliable data dissemination protocol for wireless sensor networks, Wireless Sensor Networks (2008) 268–285.

[25] K. Lin, P. Levis, Data discovery and dissemination with dip, Proceedings of the 7th international conference on Information processing in sensor networks (2008) 433–444.

[26] T. Dang, N. Bulusu, W.-c. Feng, S. Park, Dhv: A code consistency maintenance protocol for multi-hop wireless sensor networks, Wireless Sensor Networks 5432 (2009) 327–342.

[27] P. Levis, N. Patel, D. Culler, S. Shenker, Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks, Proceedings of the 1st USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI) (2004) 15–28.

[28] G. Fersi, W. Louati, M. B. Jemaa, Distributed hash table-based routing and data management in wireless sensor networks: a survey, Wireless networks 19 (2) (2013) 219–236.

[29] S. A. Abid, M. Othman, N. Shah, A survey on dht-based routing for large-scale mobile ad hoc networks, ACM Computing Surveys (CSUR) 47 (2) (2015) 20.

[30] M. Caleffi, L. Paura, M-dart: multi-path dynamic address routing, Wireless Communications and Mobile Computing 11 (3) (2011) 392–409.

[31] S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, D. Estrin, Data-centric storage in sensornets, ACM SIGCOMM Computer Communication Review 33 (1) (2003) 137–142.

[32] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, F. Yu, Data-centric storage in sensornets with GHT, a geographic hash table, Mobile Networks and Applications 8 (4) (2003) 427–442.

[33] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, S. Shenker, GHT: a geographic hash table for data-centric storage, Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications (2002) 78–87.

[34] B. Karp, H. Kung, GPSR: Greedy perimeter stateless routing for wireless networks, Proceedings of the 6th annual international conference on Mobile computing and networking (2000) 243–254.

[35] M. Ali, Z. Uzmi, CSN: A Network Protocol for Serving Dynamic Queries in Large-Scale Wireless Sensor Networks, 2nd Annual Conference on Communication Networks and Services Research (CNSR'04) (2004) 165–174.

[36] I. Stoica, R. Morris, D. Karger, M. Kaashoek, H. Balakrishnan, Chord: A scalable peer-to-peer lookup service for internet applications, Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications (2001) 149–160.

[37] M. Ali, K. Langendoen, A Case for Peer-to-Peer Network Overlays in Sensor Networks, Proceedings of WWSNA / 6th IPSN'07, Cambridge, MA 6.

[38] H. Pucha, S. Das, Y. Hu, Ekta: An Efficient DHT Substrate for Distributed Applications in Mobile Ad Hoc Networks, Proc. 6th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA) (2004) 163–173.

[39] A. Rowstron, P. Druschel, Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems, Lecture Notes In Computer Science 2218 (2001) 329–350.

[40] D. Johnson, D. Maltz, Dynamic Source Routing in Ad Hoc Wireless Networks, Kluwer International Series in Engineering and Computer Science (1996) 153–179.

[41] C. Perkins, E. Royer, Ad-hoc on-demand distance vector routing, Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications 2 (1999) 90–100.

[42] T. Zahn, J. Schiller, MADPastry: A DHT Substrate for Practicably Sized MANETs, Proc. of ASWN.

[43] C. Zheng, G. Shen, S. Li, S. Shenker, Distributed Segment Tree: Support of Range Query and Cover Query over DHT, Proc. IPTPS.

[44] A. Ghose, J. Grossklags, J. Chuang, Resilient Data-Centric Storage in Wireless Ad-Hoc Sensor Networks, Lecture Notes in Computer Science (2003) 45–62.

[45] H. Wirtz, T. Heer, R. Hummen, K. Wehrle, Mesh-dht: A locality-based distributed look-up structure for wireless mesh networks, IEEE International Conference on Communications (ICC) (2012) 653–658.

[46] O. Landsiedel, K. Lehmann, K. Wehrle, T-DHT: Topology-Based Distributed Hash Tables, Proceedings of the Fifth IEEE International Conference on Peer-to-Peer Computing (P2P'05) (2005) 143–144.

[47] J. Newsome, D. Song, GEM: Graph EMbedding for routing and data-centric storage in sensor networks without geographic information, Proceedings of the 1st international conference on Embedded networked sensor systems (2003) 76–88.

[48] M. Caesar, M. Castro, E. Nightingale, G. O'Shea, A. Rowstron, Virtual ring routing: network routing inspired by DHTs, Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications 36 (4) (2006) 351–362.

[49] A. Awad, R. German, F. Dressler, P2P-based routing and data management using the virtual cord protocol (VCP), Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing (2008) 443–444.
[50] A. Awad, R. German, F. Dressler, Exploiting virtual coordinates for improved routing performance in sensor networks, IEEE Transactions on Mobile Computing 10 (9) (2011) 1214–1226.
[51] A. Al-Mamou, H. Labiod, ScatterPastry: An Overlay Routing Using a DHT over Wireless Sensor Networks, The 2007 International Conference on Intelligent Pervasive Computing, IPC. (2007) 274–279.
[52] C. Perkins, P. Bhagwat, Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers, ACM SIGCOMM Computer Communication Review 24 (4) (1994) 234–244.
[53] S. A. Abid, M. Othman, N. Shah, 3d p2p overlay over manets, Computer Networks 64 (2014) 89–111.
[54] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, L. Viennot, Optimized link state routing protocol for ad hoc networks, Proceedings of the IEEE International Multi Topic Conference INMIC. Technology for the 21st Century. (2001) 62–68.
[55] S. Shin, U. Lee, F. Dressler, H. Yoon, Motion-mix dht for wireless mobile networks, IEEE Transactions on Mobile Computing 15 (12) (2016) 3100–3113.
[56] H. Barjini, M. Othman, H. Ibrahim, N. Udzir, Shortcoming, problems and analytical comparison for flooding-based search techniques in unstructured P2P networks, Peer-to-Peer Networking and Applications (2011) 1–13.
[57] C. Gkantsidis, M. Mihail, A. Saberi, Random walks in peer-to-peer networks: algorithms and evaluation, Performance Evaluation 63 (3) (2006) 241–263.
[58] X. Liu, Q. Huang, Y. Zhang, Combs, needles, haystacks: balancing push and pull for discovery in large-scale sensor networks, Proceedings of the 2nd international conference on Embedded networked sensor systems (2004) 122–133.
[59] A. Dimakis, V. Prabhakaran, K. Ramchandran, Ubiquitous access to distributed data in large-scale sensor networks through decentralized erasure codes, Fourth International Symposium on Information Processing in Sensor Networks, IPSN 2005. (2005) 111–117.
[60] A. Dimakis, V. Prabhakaran, K. Ramchandran, Decentralized erasure codes for distributed networked storage, IEEE/ACM Transactions on Networking (TON) 14 (2006) 2809–2816.
[61] K. K. Rachuri, C. S. R. Murthy, Energy Efficient and Scalable Search in Dense Wireless Sensor Networks, IEEE Trans. Comput. 58 (2009) 812–826.
[62] N. Sadagopan, B. Krishnamachari, A. Helmy, Active query forwarding in sensor networks, Ad Hoc Networks 3 (1) (2005) 91–113.
[63] K. Rachuri, C. Siva Ram Murthy, Energy efficient and low latency biased walk techniques for search in wireless sensor networks, Journal of Parallel and Distributed Computing.
[64] B. Bloom, Space/time trade-offs in hash coding with allowable errors, Communications of the ACM 13 (7) (1970) 422–426.
[65] D. Guo, Y. He, P. Yang, Receiver-oriented design of Bloom filters for data-centric routing, Computer Networks 54 (1) (2010) 165–174.
[66] P. Hebden, A. Pearce, Data-centric routing using bloom filters in wireless sensor networks, Fourth International Conference on Intelligent Sensing and Information Processing, ICISIP 2006. (2006) 72–77.
[67] X. Li, J. Wu, J. Xu, Hint-based routing in WSNs using scope decay bloom filters, International Workshop on Networking, Architecture, and Storages (IWNAS'06) (2006) 111–118.
[68] S. Rhea, J. Kubiatowicz, Probabilistic location and routing, INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE 3 (2002) 1248–1257.
[69] A. Kumar, J. Xu, E. Zegura, Efficient and scalable query routing for unstructured peer-to-peer networks, INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE 2 (2005) 1162–1173.
[70] K. S. Bok, D. W. Kwak, J. S. Yoo, A resource discovery with data dissemination over unstructured mobile p2p networks., KSII Transactions on Internet & Information Systems 6 (3).
[71] S. He, X. Li, J. Chen, P. Cheng, Y. Sun, D. Simplot-Ryl, Emd: energy-efficient p2p message dissemination in delay-tolerant wireless sensor and actor networks, IEEE Journal on Selected Areas in Communications 31 (9) (2013) 75–84.
[72] Y. Luo, O. Wolfson, B. Xu, A spatio-temporal approach to selective data dissemination in mobile peer-to-peer networks, in: Wireless and Mobile Communications, 2007. ICWMC'07. Third International Conference on, IEEE, 2007, pp. 50b–50b.
[73] A. B. Waluyo, D. Taniar, W. Rahayu, A. Aikebaier, M. Takizawa, B. Srinivasan, Mobile peer-to-peer data dissemination in wireless ad-hoc networks, Information Sciences 230 (2013) 3–20.
[74] N. Shah, D. Qian, An efficient overlay for unstructured p2p file sharing over manet using underlying cluster-based routing., KSII Transactions on Internet & Information Systems 4 (5).
[75] M. Jiang, Cluster based routing protocol,(cbrp), draft-ietf-manet-cbrp-spec-01. txt, Internet draft.
[76] B. Cohen, Incentives Build Rebustness in BitTorrent, Workshop on Economics of Peer-to-Peer systems 6 (2003) 68–72.
[77] Vuze Inc, Vuze BitTorrent Client (former Azureus), `http://azureus.sourceforge.net/`.
[78] R. Simon Carbajo, M. Huggard, C. Mc Goldrick, An End-to-End routing protocol for Peer-to-Peer communication in Wireless Sensor Networks, MiNEMA Workshop, Proceedings of the Eurosys, Glasgow, UK (2008) 5–9.
[79] R. Simon Carbajo, M. Huggard, C. McGoldrick, Opportunistic detection of relative mobility in wireless sensor networks, IEEE/IFIP Wireless Days (WD) (2010) 1–5.
[80] R. Jain, D. Chiu, W. Hawe, A quantitative measure of fairness and discrimination for resource allocation in shared computer systems, Research Report TR-301, Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA (1984) 1–38.
[81] Memsic Corporation, TelosB Datasheet, `http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html`, previously manufactured by Crossbow Technology Inc.
[82] Texas Instruments, CC2420 Chipcon 2.4 GHz Transceiver, `http://focus.ti.com/lit/ds/symlink/cc2420.pdf`.