Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com

Routing in wireless sensor networks for wind turbine monitoring

Ricardo Simon Carbajo[?]*, Esther Simon Carbajo, Biswajit Basu, Ciarán Mc Goldrick

School of Computer Science and Statistics, School of Engineering, Trinity College, Dublin, Ireland

ARTICLE INFO

Article history:
Received 19 February 2016
Received in revised form 7 April 2017
Accepted 21 April 2017
Available online xxx

Keywords:
Wireless sensor networks
Wind turbine monitoring
Routing protocol
Gradient routing

ABSTRACT

Smart wireless sensor devices are rapidly emerging as key enablers of the next evolution in wind turbine monitoring. The potential for in-situ monitoring of turbine elements, employing methodologies that are not possible with existing wired technology, make it possible to attain new levels of granularity and autonomy in the monitoring of these structures. Wireless sensor devices are limited in terms of communication by the range of their radio modules and, thus, need to form networks in order to transfer data from distant points. Routing protocols are primary enablers of such ad hoc wireless sensor networks and these require the implementation of reliable and energy-efficient mechanisms to maximize network reliability and availability. Existing routing protocols cannot be directly applied to the monitoring of wind turbines without addressing the unique context and operational characteristics of these structures in multi-hop wireless communication. This work identifies the potential effects associated with the operation, environment and structure of wind turbines in wireless sensor network multi-hop communication, and proposes and evaluates a reliable routing protocol for wireless sensor networks employed in these domains.

1. Introduction

Wind turbine generators are commonly deployed in remote environments with difficult access and changeable meteorological conditions that can affect the reliability, operation and behaviour of mechanical and structural components—such as blades, tower, nacelle, gearbox, generator, etc. Moreover, the wind turbine blades can rotate at varying velocities, producing unpredictable effects in the vibration characteristics of different parts of the wind turbines [1]. Reliable monitoring of these structures is a key step in identifying variations in their efficiency of operation, and in avoiding major failures which can result in mechanical or structural malfunctioning, breakage and significant financial loss [2,3]. The use of wireless sensors has been proposed as a solution to monitoring multiple components inside and outside these structures, principally targeting scenarios where wired instrumentation is impractical or difficult to install [4–6]. Unobtrusive wireless sensors can be embedded in the components of wind turbines so that they can monitor and wirelessly transmit a variety of parameters of the structure. Data is transferred among those sensor devices, to and from a set of sink nodes, which are capable of establishing connectivity with other networks such as the Internet. This follows the monitoring paradigm where data flows between the wireless sensor network and the centre for data processing and control. While this is the traditional monitoring paradigm, we need to consider that wireless sensors integrate memory and processing capabilities to perform computational activities and that we should leverage these resources to alleviate the burden of wireless communication [7]. In other words, we should pre-process data onboard sensor devices, or within the

* Correspondence to: School of Computer Science and Statistics, O'Reilly Institute, Trinity College, Dublin 2, Ireland. Fax: +353 16772204.
School of Computer Science and Statistics, O'Reilly Institute, Trinity College Dublin 2 Ireland
Email addresses: carbajor@tcd.ie (R.S. Carbajo); esimonca@tcd.ie (E.S. Carbajo); basub@tcd.ie (B. Basu); Ciaran.McGoldrick@scss.tcd.ie (C. Mc Goldrick)

http://dx.doi.org/10.1016/j.pmcj.2017.04.007
1574-1192/

sensor network (also known as in-network processing), to reduce the communications traffic within the wireless sensor network [8]. There are some key advantages in following such an approach:

(i) Reducing wireless communications will reduce the energy consumed by each node as the radio transceivers give rise to the most significant energy consumption.
(ii) Reducing wireless communications lessens the network traffic load and thus improves throughput, latency and reliability in data delivery.
(iii) Pre-processing data sampled at high data rates will help avoid streaming potentially redundant traffic in the wireless sensor network. Streaming data at high rates through multiple wireless sensor nodes contributes to a decrease in the network performance and quality of service and can provoke a deadlock situation—particularly if multiple nodes try to stream at the same time. This can occur when monitored parameters, such as vibration or wind speed, vary rapidly.

To the best of our knowledge, a routing protocol for networking of wireless sensor nodes in wind turbines, which takes into account the requirements posed by the behaviour of such structures, has not been described in the literature. The design of such a routing protocol for wind turbine monitoring will demand an informed solution providing reliable (i) Point-to-Point (P2P), (ii) Point-to-Multipoint (P2M), and (iii) Multipoint-to-point (M2P) communication within the WSN, for system configuration, in-network processing, and data transfer purposes. Reliability, and timeliness, in the delivery of data, such that data loss is ameliorated, are amongst the main considerations and features that the routing protocol needs to provide. Whilst reliable mechanisms are required, the correct and appropriate operation of the network will also depend on the managed access of sensor devices to the wireless medium, which in turn is influenced by the level of pre-processing of data at the application layer. An admission control mechanism, integrated within the routing layer, can help control the rate at which applications can inject data into the network. The primary considerations arising in interconnecting WSN's deployed in, or on, wind turbine structures are now explored.

(1) The rotational speed of Wind turbine blades varies and their pitch can also be feathered and locked, e.g. when the wind reaches excessive velocities. Sensor devices deployed at the blades will be rotating and consequently their neighbourhood (contactable nodes) will be changing, e.g. when the blade gets closer to the tower where sensors are also deployed. While opportunistic communication can occur when within appropriate range of a new node, appropriate identification of core persistent sensor nodes in the neighbourhood of a rotating node is fundamental to improving reliability, e.g. nodes along the blade should act as routers for nodes at the tips of the blade.
(2) The protocol needs to adapt to change, e.g. when a new node is incorporated to the network, by identifying the persistent neighbours of each node and maintaining the network in as fully connected a configuration as possible. Asymmetric links, both transient and persistent, must be accommodated. These may result from motion, and from interference generated from vibrations, weather conditions and other external factors. Moreover the wind turbine components may themselves be constructed from signal attenuating materials.
(3) Wind turbine generators are connected via backbone (wired) networks to the data analysis and control centre. The most straightforward interconnection of the WSN to the backbone network is to have a wireless sensor node acting as gateway [6]. However, reliability and availability remain key concerns and multiple nodes should be capable of serving as gateways. The protocol should enable this type of configuration, where any node can act as a transient or persistent gateway, and also enable bidirectional communication. This functionality will be specifically useful in those cases where in-situ maintenance activities may require integration into the network at any point, i.e. through any sensor device in range. Thus, sensors should be self-configuring and self-describing when connected, and be queryable in a human-readable manner, to facilitate their identification to the network and operators.
(4) Maintenance inspection of wind turbines is expensive and such activities do not occur overly frequently [2]. Consequently, sensors must manage their energy source to remain operational for extended periods. Minimizing communication activities is one of the main strategies for reducing energy consumption; nonetheless sensing and data processing can also deplete the battery in just a few days. Where possible, the application layer should put the sensor to sleep in order to conserve energy, whilst not impeding the maintenance of maximal network connectivity. The routing layer needs to trade-off the availability of the node to route data, with the consumption of energy associated with the activities. For these purposes, a dynamic sleeping mechanism needs to be in place which reacts to the network traffic status in the vicinity of each node.

Having regard to the challenges identified above, when performing multi-hop communication in wireless sensor networks deployed along wind turbine structures, a novel routing protocol has been developed. The routing protocol, known as the Ubiquitous Mobile Gradient(UMG), employs a reactive gradient-based routing paradigm. The novelty of the UMG routing protocol resides in the communications versatility provided and the integrated collection of mechanisms which enhance the efficiency and reliability of the routing process. In this sense, UMG:

- provides reliable point-to-multipoint, multipoint-to-point and point-to-point communications whilst following the reactive paradigm,

- combines address-centric and data-centric routing concepts to provide service advertisement and discovery to higher layers,
- employs a backoff-based reliable controlled flooding mechanism for the progressive formation of gradients according to the hop distance,
- allows for the coexistence of multiple gradient updates from the same sink node, thereby facilitating the participation of nodes which were unable to update in the forwarding process whilst avoiding loop formation,
- provides a mechanism to aggregate nodes in "gradient clusters" and to define inter-cluster gradient formation rules,
- employs an efficient short memory cache to establish reliability when descending the gradient, and to identify which end-to-end acknowledgement packets are to be broadcast when ascending the gradient with the goal of providing end-to-end reliability,
- integrates an opportunistic relative mobility detection mechanism which enables the discovery of persistent and transient nodes in the neighbourhood of a node.

The remainder of this paper is organized as follows: The next section provides a comprehensive review of routing protocols for wireless sensor networking with an emphasis on gradient-based routing protocols. A description of the UMG routing protocol then follows, and incorporates the rationale for the key features which drive its design in the context of WSNs for wind turbine monitoring. The following section describes the protocol operation and places the main phases of the protocol into context. There then follows a detailed description of the design and implementation of the Gradient Spread phase of UMG. Next, a mechanism for service description, advertisement and discovery is presented as an integral element of the protocol. The design and implementation of the three data transport phases: Gradient Descent, Local Repair and End-to-End Acknowledgement, are described in Section 3.4. Thereafter, a mechanism which deals with mobility of sensor nodes in wind turbine structures is presented. Subsequently, a traffic-oriented dynamic sleeping policy for UMG is described which enables nodes to save energy while still participating in the routing process. The set of experiments performed, comparing UMG with CTP, the de-facto routing protocol for collection in WSNs, is then set out, and UMG is evaluated under different scenarios to demonstrate its reliability and suitability for wind turbine monitoring. Finally, this work is concluded and directions for future work are indicated.

2. Related work on gradient-based routing protocols for WSNs

Many routing protocols for wireless sensor networks have evolved from routing techniques in wireless networks and have been adapted to suit the requirements and constraints of WSNs. Flooding and Gossipping [9] are two of the most commonly employed techniques for dissemination and searching in ad hoc wireless networks and in WSNs. These approaches can be classified [10] according to: (a) the Network Structure, (b) the Route Discovery Process (Reactive, Proactive and Hybrid) and (c) the Protocol Operation. For instance, reactive routing protocols, like Ad hoc On-Demand Distance Vector (AODV) [11] and Dynamic Source Routing (DSR) [12], have been adapted to work on an on-demand basis: communicating only when there is data in the network to be transmitted. The reactive approach is employed in UMG in order to avoid the energy-greedy, periodic beacons used by proactive protocols, such as the Destination-Sequenced Distance-Vector (DSDV) [13].

An alternative classification modality and communication paradigm arises when we take account of the large volume of devices, which may overlap in the monitoring of an area or phenomenon. Herein data and metadata become more important than the devices which generate it. This concept is referred to as "data-centric" and was introduced in a protocol called Directed-Diffusion [14]. "Data-centric" is different from the more traditional "address-centric" approach, where the address of the device is the primary identifier in the communication process. Data-centric routing protocols need to be aware of the type of application data, and the network characteristics underpinning the design of the routing mechanism. Many protocols employing "data-centric" routing are also classified as data dissemination protocols, where the main goal is to push data through the network towards a sink, or set of sinks, in a process where intermediate nodes do not need to be known and act as relays. Data-centric Routing Protocols for data dissemination include Directed Diffusion [14], SPIN [15], Rumour Routing [16] and Gradient-Based Routing (GBR) [17]. The design of UMG includes both the address and data centric approaches.

These classifications establish points of reference for defining the behaviour of a routing protocol; however most of the routing protocols can be associated with more than one type of each proposed classification. Moreover, features like mobility tolerance, energy awareness and reliability are of significance in WSN environments and have to be considered in their design. One of the main targets of this work is to build a generic routing protocol capable of adapting to multiple network scenarios and different applications.

2.1. Concepts of gradient-based routing in wireless networks

The UMG protocol employs the "gradient-based" routing paradigm. Gradient-based routing protocols are employed as mechanisms for data collection in multi-hop mesh networks where a source node, i.e. the sink, spreads its gradient through the nodes in the network in a process known as gradient setup. The gradient is formed by broadcasting packets in such a way that every node receiving a packet, updates its routing table and (re)broadcasts the packet. This process creates a Directed Acyclic Graph (DAG) where routing towards the root node is achieved by following the direction of the graph in a process called "gradient descent", reducing the height to the sink in each routing hop.

The concept of sending data towards a central node (with higher capabilities) is frequently employed in ad hoc networks for network data processing or for the purposes of pushing data outside of the network. Many of these gradient-based protocols employ a unidirectional approach where data is routed from nodes to only one sink node. Some employ clustering algorithms to designate the group of nodes belonging to the sink gradient overlay network. In multiple sink protocols mechanisms to evaluate the current network performance and available resources are employed to optimize the routing process and balance the network traffic.

Some of the more important gradient routing protocols are now assessed in terms of their capabilities and operation.

2.1.1. Review of gradient-based routing protocols

One of the first gradient based approaches in ad hoc wireless networks was "Gradient Routing in Ad Hoc Networks" [18] in 2000. "GRAd" belongs to the reactive category of ad hoc routing protocols where routing information is established on-demand. In this approach, the general concept is "if you want to be spoken to, you must first speak". This protocol employs opportunistic update of routing cost tables. Around the same time, the first WSN protocols using gradient and data centric routing concepts for data dissemination were presented as "Directed Diffusion" (DD) [19,14] and Gradient Based Routing (GBR) [17]. The GBR routing protocol presumes that optimal routing in sensor networks is infeasible. A special packet called "Interest" is employed to setup the gradient, employing the data centric approach, while a dedicated packet is used to transport the data. Nodes satisfying the requirements in the "Interest" packet send their "Data" packets to the sink by descending the established gradient. This approach enables energy efficient techniques like in-network data processing with data combination.

In 2001, Ye et al. proposed a scalable solution to minimum cost forwarding in large sensor networks [20]. The algorithm constructs an efficient gradient in terms of forwarding cost which employs a controlled flooding where sensors only broadcast each unique gradient discovery packet once. An enhanced version of their protocol was presented later as Gradient Broadcast (GRAB) [21,22], which provides a robust mechanism for data forwarding when descending the gradient. The original gradient setup mechanism, where the energy consumed in each link is employed as the cost value, is retained [20], and the gradient is also refreshed based on the variation of each sources historic profile. Subsequently, three optimizations of the forwarding phase of GRAB were proposed as "Probabilistic-GRAB" [23], "Utility-GRAB" [24] and "Utility and Probabilistic GRAB" [25], which take into account interference and congestion metrics. In the same vein, Xia et al. combined the hop count and the remaining energy at each node to calculate the cost value when creating the gradient in order to prolong network lifetime [26].

GLIDER [27] is a gradient routing protocol where each node contains a lightweight global vision of the routing information. The scheme assumes that nodes in a sensor network can disappear but the main global topology is maintained. The network is partitioned into tiles which are represented by a landmark. The protocol creates an overlay of landmarks, which are nodes acting as reference points for its tiles using the landmark Voronoi complex., i.e. regions of nodes with a landmark node virtually representing it.

Other works focussed on developing mechanisms for improving the gradient-ascending process. For instance, GRASP [28] operates in stationary networks where the gradient has already been established. It makes use of data packets which descend the gradient towards the sink to populate a Bloom filter [29,30], a probabilistic bit-vector structure which acts as a membership-based filter capable of indicating whether the packet has been stored or not.

While the majority of these protocols work in static networks, other approaches offer solutions for dealing with mobile nodes. As a collection routing protocol, Hyper [31] offers support for mobile sinks. The protocol can collect data at any given time from any given point in the WSN. The sink node advertises its arrival to the neighbourhood and evaluates the link quality to each of its sensors with a "fast connect" mechanism. The Reliable Cost-based Data-centric Routing Protocol for Wireless Sensor Networks (RCDR) [32] routes data from an event zone towards the sink in dynamic environments where the sink node can change position.

Employing energy as the cost metric, in State-free Gradient-based Forwarding (SFG) [33], the gradient is established at the beginning of the network and it is updated via received data packets. Ye et al. [20] improved this scheme by employing a back-off-based gradient formation mechanism. The gradient cost value is defined as the minimum total energy required to send a packet from the node itself to the sink; this takes account of energy consumed at every node in both the transmission and reception of the packet. Khan et al. [34] present "GRAdient cost establishment (GRACE) for an energy-aware routing in wireless sensor networks" which enhances network lifetime and reliable data delivery when compared to GRAB. Guo et al. presented Dynamic Gradient-based Routing protocol [35] which aims to balance the energy consumption by detecting nodes at certain energy levels and refreshing the gradient with new cost values. DGR has been developed based on concepts from SGF [33] and GRAB [21,22] protocols. The number of nodes involved in the routing process, known as the "expansion strategy", is controlled by a parameter in the packet. The protocol assumes that nodes can change the communication range by varying the transmission power at any time.

The challenge of increasing reliability in the presence of lossy links was addressed by TABS (Try Ancestors Before Spreading) [36]. When descending the gradient towards the sink, a sender broadcasts a packet which contains a value known as the "minimum progress limit". Receiving nodes compare their cost with that of the received packet and keep on broadcasting if the difference is higher or equal to the minimum progress limit. TABS controls nodes involved in the forwarding process in the same way GRAB [21,22] controls the width of the forwarding band.

The Collection Tree Protocol (CTP) [37] is the de-facto standard in data collection protocols for Wireless Sensor Networks on the popular platforms TinyOS 2.x [38] and Java SunSPOTS [39]. CTP is a tree-based protocol where a designated root, or set of roots, advertise themselves and create routing trees by spreading their gradients. Root nodes establish their gradient using a cost metric called Expected Transmission Count (ETX). The ETX metric for each node estimates the link quality based on the number of successfully delivered unicast packets, i.e. when an acknowledgement packet is received, between two given nodes. The ETX of a node is the ETX value received from its parent plus the ETX cost of the link from the node to its parent. The ETX at the root is 0 when the gradient is setup. This measure helps to avoid loops when data packets descend the gradient towards the root by always progressing through a node with lower ETX. When a loop is detected, an inconsistency in CTP terms, the node broadcasts a beacon to inform the sending node that it needs to adjust its routes. The parent node reacts to inconsistencies by sending a beacon frame to all its neighbours to update them. In the case where the network is partitioned, legitimate duplicate packets, for instance those which circulate through the loop twice, need to be detected but not discarded. To detect this type of packet, a counter named "time has lived" (THL) is carried in the data packets.

Two coupled mechanisms are employed to discover bidirectional links; every 5 transmissions the ETX value is computed based on the previous ETX and the successful packet delivery ratio. The first mechanism, called LEEP, employs beacon messages containing in-bound link estimations (from transmissions by nodes in the neighbourhood) [40]. Nodes receiving the beacon can thus calculate their out-bound link estimation. The Trickle dissemination protocol [41] self-regulates the beacon update period based on the changes in the neighbourhood according to parameters contained in the packets. In the case of CTP, an inconsistency or significant decrease in its routing cost will reset the period of beacon transmission to a minimum value. The minimum beacon transmission values can be of the order of 60 ms with maximum values of the order of 1+ hours. In addition, ETX values can be updated by unicast data packets and associated acknowledgement packets.

Gnawali et al. developed CTP Noe [42], which includes additional functionality to improve the protocol's performance. CTP Noe uses retransmit timers of up to 32 trials and its beacon rate in Trickle has been set to a minimum value of 64 ms and a maximum value of 1 h. Changes in the topology (mainly link dynamics) can be detected rapidly, while at the same time avoiding redundant communication when the network is stable. In addition, CTP Noe employs a 4-bit link estimator metric as a composite value of the quality of the link from the physical, mac, and routing layer [43]. CTP Noe offers 90%–99.9% packet delivery in highly dynamic link topologies with 73% fewer control packets than existing approaches. The authors claim it can achieve duty cycles of less than 3% while supporting aggregate loads of 25 packets per minute.

A variety of protocols have been inspired by the Collection Tree Protocol (CTP) [37]. The Backpressure Collection Protocol (BCP) [44] implements the backpressure routing concept for routing packets in multi-hop networks. This is achieved by employing queue congestion status in the forwarding of a packet. BCP needs to initially create a gradient, based on the ETX metric, similar to CTP [37], where routing trees are created with the root node as the sink collector. The Whirlpool Routing Protocol (WARP) [45] is an extension of CTP which includes an enhancement for routing to mobile sinks. A sink node in WARP detects its mobility by using periodic beacon messages.

In 2010, the IETF Routing Over Low power and Lossy networks (ROLL) Working Group was formed [46] to standardize a generic routing protocol which satisfies most of the requirements and constraints from scenarios where low power and lossy networks are to be deployed [47–50]. An IPv6-compliant gradient-based routing protocol, known as RPL (Routing Protocol for Low power and lossy networks) [51], has been engineered to route data to thousands of low power devices where interconnections are characterized by high loss rates, low data rates, and instability. The protocol design employs the gradient based concept and describes the basis of routing such as packet processing and forwarding decisions while leaving the network traffic optimization mechanisms open for implementation by the application developer. The IETF ROLL group work indicated the necessity for the routing protocol to support point-to-point (P2P), point-to-multipoint (P2MP), and multipoint-to-point (MP2P) communication, to account for all possible traffic flow application scenarios.

A targeted review of gradient-based routing protocols for WSNs has been presented. Common features of gradient-based routing protocols, including those of Ubiquitous Mobile Gradient (UMG), have been identified and are presented in Table 1. The table helps to characterize the behaviour and functionality of UMG when compared to existing protocols. UMG's design can be considered alongside DGR [35] or GRACE [34] as they have some commonality of core feature set, including a reactive behaviour, unicast-based gradient descent and gradient ascent functionality. However, UMG supports sink mobility and does not incorporate energy in the routing metric.

While having some commonalities with existing gradient protocols, UMG's design integrates a set of specific mechanisms to enhance the reliability and versatility of the routing process. UMG incorporates a robust gradient formation mechanism based on a backoff delay as a function of the hop count. It employs a mechanism which avoids cycles and allows for the coexistence of cost table information from different gradient formation processes from the same sink node. UMG employs reliable end-to-end mechanisms utilizing a unicast-based gradient descent phase to transport data, while using broadcast packets for acknowledgement purposes when ascending. UMG also employs a novel mechanism to detect relative mobility for triggering scoped sink gradient updates, and leverages the routing process to provide service advertisement to higher layers.

Table 1
Comparison of gradient-based routing protocols.

	Reac	Proac	Asym	Cost	Gradient creation	Height update	Descent	Ascent	Mobi
GRAd [18]	✓			Hops	Flood, 1stPck	Setup	Broadcast	✓	
GBR [17]	✓	✓		Hops	Flood, 1stPck	Neigh	Unicast		
A scalable… [20]	✓			EnergyComms	Flood, BestPck, Delay	Setup	Unicast		
GRAB [21,22]	✓		✓	EnergyComms	Flood, BestPck, Delay	Setup	Broadcast		
GLIDER [27]		✓		Hops	N/A	N/A	Unicast		
A new… [26]	✓		✓	Hops+Energy	Flood, BestPck, Delay	Setup	Broadcast		✓
GRASP [28]	✓			Hops	N/A	N/A	N/A	✓	
HYPER [31]		✓	✓	ETX+LQI	Flood, 1stPck, Delay	Beacon	Unicast		✓
RCDR [32]	✓		✓	EnergyComms	Flood, BestPck, Delay	Setup	Broadcast		✓
SGF [33]	✓	✓		EnergyComms	Flood, BestPck, Delay	Setup, Data	Unicast		
GRACE [34]	✓		✓	EnergyComms	Flood, BestPck, Delay	Setup, Ack	Unicast	✓	
CTP [37]		✓	✓	ETX	Flood, 1stPck	Beacon, Data, Ack	Unicast		
BCP [44]		✓	✓	ETX+QB	Flood, 1stPck	Beacon, Data	Unicast		
WARP [45]		✓	✓	ETX	Flood, 1stPck	Beacon, Data, Ack	Unicast		✓
TABS [36]	✓	✓	✓	Hops	N/A	Setup	Broadcast		
DGR [35]	✓		✓	Hops+Energy	Flood, 1stPck	Setup	Unicast	✓	
RPL [51]		✓	✓	User-defined	Flood, 1stPck	Beacon	Unicast	✓	
UMG	✓		✓	Hops	Flood, 1stPck, Delay	Setup, Ack	Unicast	✓	✓

Reac/Proac: Reactive/Proactive Route Discovery, **Asym**: Asymmetric Link Tolerance, **Cost**: Routing Metric for Gradient Creation and Navigation (EnergyComm: Energy Radio Transmission; Energy: Device Energy; QB: Queue Backlog), **Gradient Creation**: Gradient Creation Mechanism (Flood: Flooding; 1stPck: 1st Packet Received is Broadcast; BestPck: Best Cost Packet is Broadcast; Delay: Back-off Delay before Pck is Broadcast), **Height Update**: Cost Table Updated (Setup: In the Gradient Setup Process; Beacon: With Periodic Beacons; Neigh: With Explicit Neighbour Pck; Data: With Data Packets; Ack: With Ack Packets), **Descent**: Gradient Descent (Broadcast: Data Pck is Broadcast according to Threshold with Cost Decrement; Unicast: Each Router Selects Next Forwarder following the Routing Table or Local Repair), **Ascent**: Gradient Ascent—Ack Pck from Sink to Source, **Mob**: Sink Mobility Tolerance.

3. UMG: gradient-based routing in WSNs for wind turbine monitoring

A dynamic gradient-based reactive routing protocol, known as Ubiquitous Mobile Gradient (UMG), has been designed to operate in Wireless Sensor Networks deployed in Wind Turbine structures. Its design supports efficient and reliable communication for a variety of application scenarios in the area of WSNs monitoring and control.

UMG makes use of the gradient concept which describes a special type of routing employing an overlay network in which every node has an associated quantity value with respect to a node known as "sink". Routing of packets is achieved by decreasing the quantity value at the next node in the path while descending the gradient towards the sink node, i.e. relaying the packet to those neighbour nodes which have lower quantity values with respect to the sink. The sink needs to spread its gradient field through the nodes in the network to create a Directed Acyclic Graph (DAG) where every participating node has an associated value with respect to the root node, i.e. the sink, which is ultimately reached by following the direction of the graph in a process called "gradient descent".

UMG offers reliable mechanisms for the creation, update and navigation of the gradient field and supports point-to-point, multipoint-to-point and point-to-multipoint communication. The protocol has been designed to operate in networks where every node can take the role of consumer and/or producer. In this way nodes can be sinks and sources at the same time. 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 [29] 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. Services range from sensed or fused data to resources available at the node or a description of the node itself. In this sense, UMG acts as a service advertisement protocol and facilitates the data-centric searching process. The potential for each node to describe itself allows for applications to create multiple overlays according to node's services or node's interests in data, thereby making possible the formation of dynamically changing communities, or clusters, of wireless sensor nodes. Each node might belong to multiple communities indicated by the node's descriptor acting as the node's profile.

One of the goals of UMG is to avoid the use of periodic messages, assuming that areas with no data communication are not updated, whilst network areas with activity benefit from the opportunistic communication in order to maintain its connectivity status up-to-date. In other words, UMG follows the reactive paradigm, employing eavesdropping to update neighbours rather than using periodic updates. This contributes to energy saving and enables the development of traffic-oriented dynamic sleeping strategies where nodes sleep according to the activity in the wireless medium.

UMG also tolerates sink and source mobility by opportunistically detecting a node's relative change in neighbourhood, and react to changes by locally updating its gradient. This mechanism has been described in previous works [52] and operates as a standalone component identifying the core neighbourhood of a node from those nodes which exhibit a transient behaviour. However, a different and specific mobility paradigm arises when monitoring wind turbine structures where (i) the deployment of devices is supervised and (ii) the mobility of the set of nodes placed at the blades exhibit a semi-deterministic cyclic mobility behaviour. Taking this into account, UMG integrates a tailored mechanism which deals with the mobility of nodes in a more efficient and reliable manner. The mechanism enables the application layer to enhance the routing process by defining special grouped areas or flat-topology clusters, as well as their basic interaction. For instance in a wind turbine structure a cluster for each blade, a further cluster for the nacelle, and another for the tower cluster define the main groupings. In this way, the gradient formation process can be supervised and optimized to adapt to the behaviour of the structure.

3.1. Operation of UMG

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 minima and inefficient path lengths. In UMG, a new node in the network might decide at some stage to spread its gradient to create routes from other nodes towards itself. The node spreads its gradient either because it has taken a producer/consumer role in the network or because another node requests to contact the node. In the reminder of the paper, a node spreading its gradient at any given time will be known as "gradient origin node" or "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 of 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, i.e. the "Gradient Spread Phase". 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. This way, the gradient spreading process is also used as a service advertisement mechanism. Only the first gradient packet is forwarded from those received for a given gradient process from a gradient origin node. The use of the lowest hop count metric can easily be incorporated in UMG to create the shortest path; this is possible as 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. However, UMG employs the fastest route metric by default as: (i) it produces good results in terms of hops, despite not always being optimal, (ii) it requires low complexity, and (iii) it inherently considers 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 broad-

cast such that the implosion problem is minimized. For this purpose, and in order to avoid loops, duplicate packets are detected. In the same line, the broadcast of a packet might get lost due to contention problems or inactivity of the node, e.g. due to duty cycling. In this case, the gradient might not get formed, for instance, if a packet does not reach a crucial node, i.e. a node which, by its unique position, serves as the gateway in between areas of the network. For this situation, UMG adds a delayed extra broadcast retransmission of the first gradient packet in order to increase the reliability of the gradient formation.

A node receiving the gradient packet creates or updates the entry for the gradient origin node in its routing table. The entry in the routing table contains the neighbour address from which the packet was received. The routing table is composed of one entry for every gradient origin node. An entry in the routing table for the gradient origin node indicates the next node address to reach the gradient origin node, the service descriptor of the gradient origin node and the distance in number of hops to the gradient origin node. The service descriptor is stored in the form of a Bloom filter which compresses descriptive information efficiently and can also be efficiently and quickly queried (see Section 3.3). The routing protocol provides functionality to search and compare descriptors. When combining descriptors with information on the distance to the gradient origin node, the UMG offers powerful information to higher layers for the decision making process.

Once initial gradients are established, communication with the gradient origin node can be started from any other participating node in the gradient. This is achieved by descending the gradient in a reliable manner, i.e. the "Gradient Descent Phase". Every data packet descending the gradient is acknowledged at each hop, either with an explicit packet or by snooping the next hop transmission. If after a maximum number of trials, the packet has not been acknowledged, a local repair mechanism is launched which looks for a valid candidate to keep on descending the gradient towards the gradient origin node, i.e. the "Local Repair Phase". If the local repair fails, as it does not find a suitable neighbour, UMG reacts according to the mode in which the data packet is configured. A data packet can be sent configured to indicate the gradient origin node, i.e. the final destination node, to issue an end-to-end acknowledgement packet. If the local repair fails to find a suitable neighbour at an intermediate node, and an end-to-end ack packet is to be issued (indicated by the data packet), 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. However, if the local repair fails after exhausting the maximum number of trials and the data packet indicates that no end-to-end ack packet is required, then the node can trigger a gradient formation process requesting the gradient origin node to spread its gradient. In this way, the gradient is repaired for both the intermediate and the initial sender of the packet and the packet is sent by the intermediate node towards the new gradient.

End-to-end acknowledgements are enabled per packet and enhance the reliability of the data delivery process. End-to-end acknowledgement packets are issued by the gradient origin node and climb the gradient towards the originator of the communication (see "End-to-End Acknowledgement Phase" in Section 3.4.3). 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 end-to-end acknowledgement packet is only broadcast by those nodes which previously participated in the sending of the data packet. This way, the end-to-end acknowledgement packet climbs the gradient on a hop-by-hop basis. The entry for a particular data packet in the cache structure becomes invalid when an end-to-end acknowledgement packet is successfully forwarded or when it times out. The protocol is aware of the successful delivery of the end-to-end acknowledgement packet by snooping the broadcast of the next node on the way up to the originator. If an end-to-end acknowledgement packet cannot be delivered to the next node, and bidirectionality is therefore not achieved, the option of sending the end-to-end 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 worst 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.

The use of UMG without end-to-end acknowledgements has resulted in better and more efficient performance in terms of delivery ratio, latency and overhead communication costs, especially in scenarios with a high degree of noise and traffic. However, the option of end-to-end acknowledgement packets can be enabled at the application layer for selected packets and it is of great assistance when this layer needs to confirm the reception of a data packet. A diagram providing an example of the operation of the main phases of UMG is depicted in Fig. 1 which also illustrates the procedure when end-to-end acknowledgements are enabled.

In the following sections, the phases of UMG are explained: the Gradient Spread phase (including an Ubiquitous Lookup mechanism for service description, advertisement and discovery) and the Data Transport phases (Gradient Descent, Local Repair and End-to-End Acknowledgement).

3.2. Gradient spread phase

The gradient spread phase takes place when a node needs to inform other nodes in the network about both its existence and the type of service provided. This phase employs a controlled flooding process to establish a gradient towards the node, i.e. the gradient origin node, such that other nodes have the possibility to route traffic towards it. If a node does not want to be contacted or it has no service to offer, the node might choose not to spread its gradient. Instead, nodes might work simply as routers without providing or consuming data. A node can spread its gradient at any given point in time. The gradient spread phase can be launched at the beginning of the mote's life, which for most of the nodes will be the beginning of the network's life. Further gra-

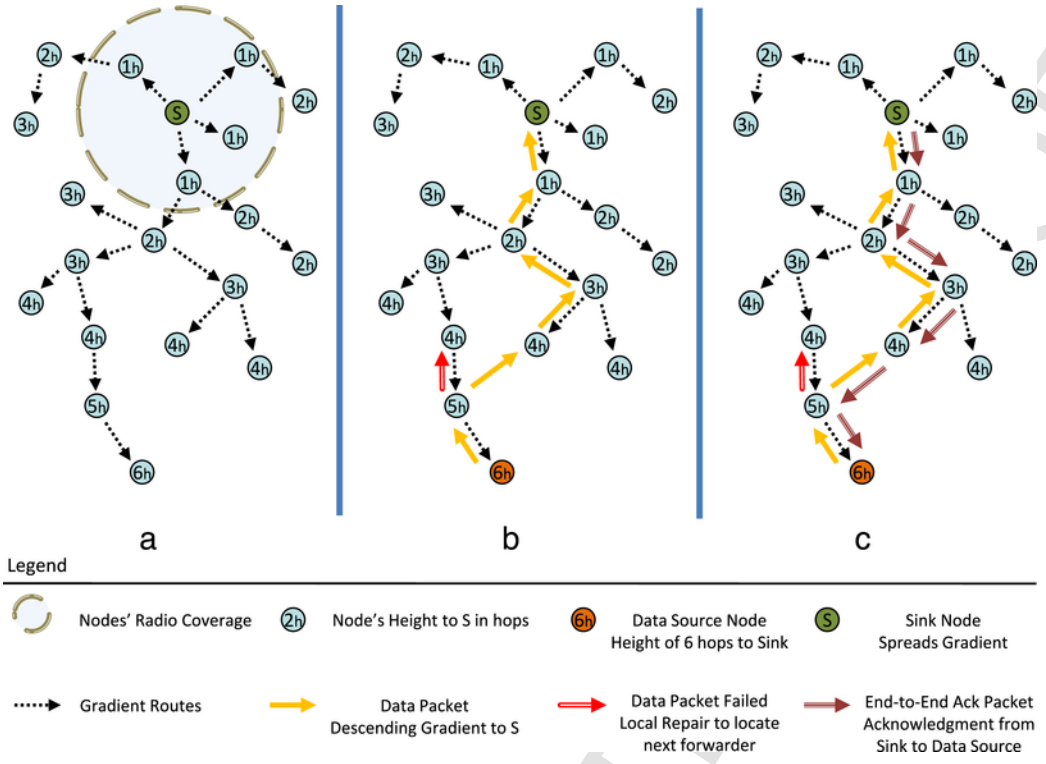

Fig. 1. UMG—Phases. Subfigure (a) Gradient Spread Phase. Subfigure (b) Gradient Descent Phase with Local Repair Phase. Subfigure (c) End-to-End Acknowledgement Phase.

dient spread updates, which might affect only a limited scope of the network, would take place when (i) a node changes its services, (ii) when reliable mechanisms demand a gradient update, or (iii) when a change in the neighbourhood of the node forces the gradient to be reconfigured. In addition, a node A has the option of requesting another node B to spread its gradient such that bidirectional data communication can occur between them; this is achieved while A is spreading its gradient.

3.2.1. Gradient formation

In the initial network setup, i.e. at the beginning of the network's life, nodes wishing to advertise their gradients wait a random time to minimize collisions with ongoing gradient processes. This process gives time to all the nodes in the network to boot up and start their radio communication system, such that they can participate in the gradient setup process. The option of using a gradient message as a control message to start the formation of gradients once all nodes have booted up is also available. In this case nodes stay on a waiting state until the first gradient packet is received from a previously agreed node or with a reserved service description encapsulated; at this time nodes in need to advertise their gradients wait for a random bounded time. While this is a proactive process where gradients are created beforehand, nodes might also setup their gradients in a reactive manner when data communication is needed.

The spreading of the gradient consists in a controlled flooding process where the first unique packet received is sent twice for a "good" formation of the gradient. The message structure employed to spread the gradient is known as "SpreadGrad" message (see Fig. 2). The address of the gradient origin node spreading the gradient is indicated in the SpreadGrad message as the "originGradAddr" field. Each node employs the SpreadGrad message to advertise its services. For this purpose, a memory efficient structure containing the service descriptor is carried in the SpreadGrad message as "descriptorBF" (see Section 3.3 for further explanation). In order to distinguish between different gradient updates from the same gradient origin node, a sequence control value is defined in the SpreadGrad message as "seq". The combination of "originGradAddr" and "seq" makes the SpreadGrad message uniquely identifiable. The sequence value is incremented each time a gradient origin node starts a gradient formation. The SpreadGrad message also contains a value which carries the height of the sender node with respect to the gradient origin node; by default UMG employs the hop distance as the height. The gradient setup limits the scope of the spreading process in terms of hop distance; this is indicated by the "maxHops" field in the SpreadGrad message.

The protocol also provides the option for a gradient origin node to request other node(s) to setup their gradient. By encapsulating requesting information in the SpreadGrad message, the gradient origin node can make requests in two ways: (i) by addressing a particular node indicating the address of the node in the SpreadGrad message as the "requestGradAddr', or (ii) by ad-

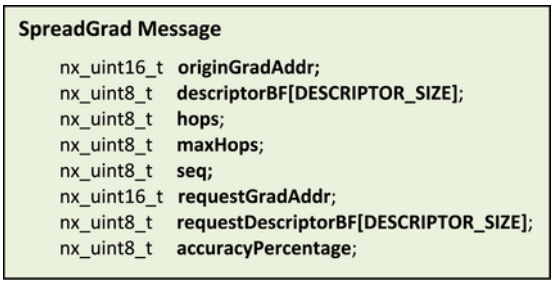

Fig. 2. UMG—SpreadGrad message structure.

dressing a set of nodes whose service descriptor matches the descriptor "requestDescriptorBF" of the SpreadGrad message with an accuracy equals or higher than "accuracyPercentage". Upon reception of the SpreadGrad message, matching nodes hold the power to decide whether to spread the gradient.

In UMG, every node receiving a SpreadGrad message only takes into account the first packet received for a given gradient origin node and sequence "seq". The first gradient message received by a node populates or updates the Routing Table (see Fig. 3) with the address of the gradient origin node as the key index (see "originGradAddr" in the SpreadGrad message and in the Routing Table). In addition, the address of the node from which the message has been received is stored in the Routing Table as the next node in the gradient to reach the gradient origin node (see "receivedFromAddr" in the Routing Table). Other values from the SpreadGrad message are also stored in the Routing Table such as the hop distance ("hops"), the sequence value ("seq"), the maximum hop distance ("maxHops") and the service descriptor ("decriptorBF"). A local counter at each node is employed to timestamp the creation or update of each Routing Table entry; the "lastTimeUsed" field in the Routing Table stores the current local counter value for the associated "originGradAddr" entry. The local counter is incremented every time the node receives a gradient update or a data packet is relayed through it. This provides an indication of the freshness of information for each gradient entry, as compared to the rest of the entries, in the event of the Routing Table gets full and a "originGradAddr" entry needs to be replaced by a new one. The "lastTimeUsed" field is also employed as a mechanism to enable or disable the Routing Table entry for a gradient. A node might be interested in cancelling its routing activity for any given gradient by temporarily disabling the entry ("lastTimeUsed" equals to 0). For instance, the node might require to save battery, ameliorate the congestion, or simply indicate that the Routing Table entry is not valid as a result of mobility or unresponsive behaviour. Finally, the "hops" field is populated when the gradient is created and acts as a control variable together with the "seq" field. However, further gradient updates, e.g. when mobility is detected, or the launching of local repair processes (see Section 3.4.2) might change the real end-to-end hop distance when descending the gradient. To account for the real end-to-end hop distance to the "originGradAddr", the "realHops" field has been added to the Routing Table. The update of this field occurs in the End-to-End Acknowledgement Phase (see Section 3.4.3).

On the formation of the gradient, routing protocols employ different mechanisms to create efficient paths in terms of hops. Many implementations employ an evaluation function which decides from a set of received packets which one is to be forwarded in the creation of the gradient. This function selects the most suitable packet based on metrics like hop count, battery level, link quality or signal strength. This process consumes memory and computational resources while demanding an extra backoff time to wait for a set of packets to be received. In UMG, the first packet arriving from the flooding process is forwarded and used for update purposes. Every node only forwards the first unique packet received for a gradient origin node. The rest of the SpreadGrad messages are discarded; this contributes to avoid cycles, decrease the communication activity and thus reduce the network contention. This is a simple mechanism which aims to reduce the complexity in the gradient process. The assumption is that creating and maintaining optimal gradients, in any terms, incurs a high use of resources at relatively low benefit in dynamic networks such as WSN. Instead, UMG exploits the creation of well-formed (sub-optimal) gradients by increasing the likelihood of all the neighbours receiving the SpreadGrad message. In this regard, the successful reception of the message by all

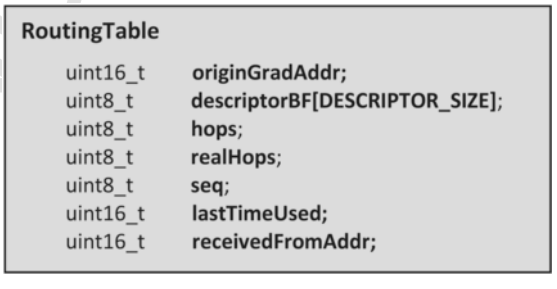

Fig. 3. UMG—Routing table.

neighbours depend on factors such as the density and topology of the network, the contention in the wireless medium, the congestion in the queues, and the nodes' sleeping policies. These factors produce packet loss and might contribute to the inefficient formation of the gradient. While the cost of not forming efficient gradients can be translated into an increase in the number of hops of a route, the true risk comes when subsequent gradient updates only reach a subset of the nodes. In this scenario, different gradient updates for the same gradient origin node need to coexist such that loops and local minima points are not produced. As an observation, gradient routing protocols which unicast data packets following the gradient path, such as UMG, are more susceptible to this problem than protocols which do not set paths and rely on nodes with lower height values to broadcast the packet.

To solve the above issues, avoid loops and add reliability and efficiency to the gradient path formation, two spaced retransmissions of the same packet are employed to increase the likelihood of all neighbours receiving the packet. This acts as a countermeasure against packet loss due to network contention or the short unavailability of a node to receive a packet. In experiments, no more than two retransmissions were required for the proper formation of the gradient even in scenarios of sparse topology with medium to high noise. Nevertheless, the gradient formation process can be re-launched for a specific area if connectivity for a particular node is not achieved. In addition, a back-off time is applied to the first and second retransmission of the broadcast packet. The back-off time is computed as a function of the hop distance to the gradient origin node, such that the delay in the retransmission increases linearly with respect to the distance from the gradient origin node. This mechanism introduces a delay in the gradient setup at the benefit of a progressive formation of the gradient in terms of distance from the origin. This can be seen as a wait and forward controlled mechanism which expands the gradient in a step-based process where each hop iteration increases the scope of the discovery progressively in a semi-uniform concentric manner with centre at the gradient origin node.

The backoff-based gradient spread mechanism works as follows: When a new unique SpreadGrad message is received for the first time at a node, an initial backoff time is calculated as:

$$\mathrm{BackOff\_Initial} = (\mathrm{hops} * \mathrm{DelayPerHop}) + \mathrm{RDDelay} \tag{1}$$

where "hops" is the number of hops received in the SpreadGrad message, "DelayPerHop" is a constant which establishes the incurred delay in milliseconds per hop unit, and "RDDelay" is a randomly generated delay in the range of [0, Delay]; "Delay" is a constant time in milliseconds. The higher the "Delay" upper limit, the higher the likelihood of an increase in the time to create the gradient and the lower the likelihood of colliding with other packets in each neighbourhood. By the same token, the higher the "DelayPerHop" value, the higher the latency in the formation of the gradient. However, the "DelayPerHop" value affects the probability of packet collision and the proper formation of the gradient. Experimentally, the probability of collision increases when the "DelayPerHop" is set to a value equal or less than 3 ms, while the probability of collision remains bounded between proximate values when the value is greater than 5 ms. Nevertheless, the probability of collision is highly dependent on the number of neighbours and the ongoing communication in the neighbourhood; these factors can be constantly changing. This is also one of the main reasons why the delay and retransmissions of the SpreadGrad message are required for the proper formation of the gradient.

Once the backoff time elapses, the packet is broadcast. A second shorter backoff time, which does not depend on the hop count, is applied to delay the retransmission of the SpreadGrad message as:

$$\mathrm{BackOff\_Retransmission} = \mathrm{RDDelay} + \mathrm{MinDelay} \tag{2}$$

where "MinDelay" is a constant which guarantees a minimum backoff time in the event of the "RDDelay" random value is zero. The BackOff_Retransmission is employed as a mechanism to increase the likelihood of all the neighbours receiving the packet. It is set to a lower value than the BackOff_Initial such that the second broadcast of the packet occurs before a packet with a higher hop value is broadcast.

While this process improves the gradient formation, it also adds a cumulative linear growth to the latency of the gradient setup, which might affect the QoS requirements of high layers. The next equation calculates the cumulative latency in the worst case scenario (where the first SpreadGrad message is lost) as:

$$\mathrm{LatencyWC} = \sum_{\mathrm{hops}=0}^{n} (\mathrm{hops} * \mathrm{DelayPerHop} + \mathrm{Delay} + \mathrm{Delay} + \mathrm{MinDelay}) \tag{3}$$

solving the series in Eq. (3), the worst case latency can be calculated as:

$$\mathrm{LatencyWC} = \frac{1}{2}(n+1)\left[(n * \mathrm{DelayPerHop}) + ((2 * \mathrm{Delay} + \mathrm{MinDelay}) * 2)\right]. \tag{4}$$

By the same token, the best case scenario, where the first packet is always received and the random value of the "RDDelay" is always 0, can be calculated as:

$$\text{LatencyBC} = \sum_{\text{hops}=0}^{n} (\text{hops} * \text{DelayPerHop}) \tag{5}$$

solving the series in Eq. (5), the best case latency can be calculated as:

$$\text{LatencyBC} = \frac{1}{2}(n+1)(n * \text{DelayPerHop}). \tag{6}$$

By default, UMG employs the next combination of values for the time delays in milliseconds: " DelayPerHop = 5ms", " Delay = 7ms", and " MinDelay = 3ms". These parameters have been empirically selected as they have proven to establish efficient gradients in terms of hop distance at low latencies for a variety of scenarios with different traffic loads. Following these values, the latency for the worst and best case scenarios is shown in Fig. 4 for a maximum hop distance of 10 hops. Note that the latencies correspond to the delay in the formation of a gradient, this does not take into account the delay introduced when other packets are in the queue, the time to send a broadcast packet (∼11 ms in TinyOS) and each packet MAC backoff time (300μs —9.8 ms in TinyOS).

Additionally, the number of messages sent and received in the gradient formation process can be estimated beforehand. The number of messages broadcast in a gradient formation process depends on the scope of the flooding and the number of nodes in the scope. However, knowing the number of participant nodes, the maximum number of messages broadcast will be 2 per node, which gives an idea of the communication impact of this process. The number of broadcast packets received at each node for a particular gradient formation depends on the network density, the transmission power range and the receiver sensitivity of the transceiver. In other words, it depends on the average number of neighbours for each node as this varies dynamically due to the instability of the wireless links. In addition, the scalability factor in UMG is directly proportional to the number of routing table entries which a node can accommodate. In other words, how many consumer/producer nodes are going to employ a given node to perform routing for their gradients. In this regard, a given node might only be responsible for a set of gradients within a limited scope. The rationale behind this is that, in large sensor networks, nodes would limit communication within a certain hop distance for efficient routing; after a certain number of hops, routing might be inefficient and infeasible.

Finally, a new node arriving to an up and running network where gradients are already formed is of no use to the network until it learns from the network on how to act as a router. The cost of re-calculating the gradients of a new node's neighbourhood to make it work as a functional router for all the existing gradients is expensive in terms of communication. Therefore, in UMG a new node will progressively operate as a router when its routing table is updated with new gradient updates. Gradient setups can be triggered due to an unresponsive end-to-end connectivity or when mobility detection demands an update. On the other hand, if a new node manifests interest in being connected to a particular node or set of nodes, the UMG routing protocol provides query mechanisms to prompt specific nodes for gradient formations.

3.3. Ubiquitous lookup

Many applications in WSN require some form of node description in terms of data or service provided. Protocols have been designed to specifically achieve such service advertisement and discovery functionality. Most of them require a high number of

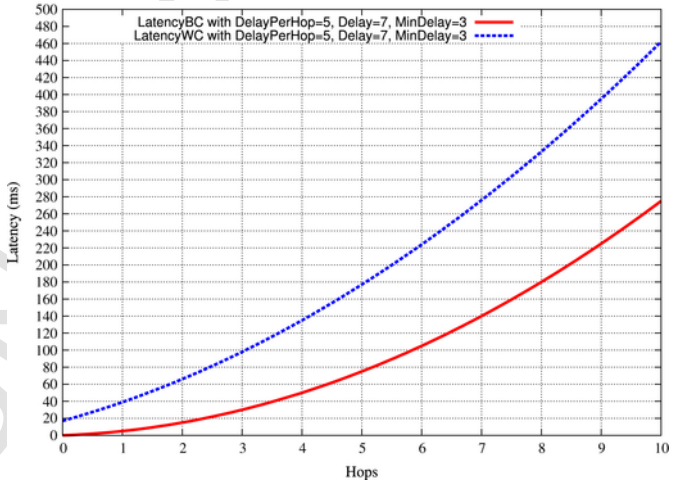

Fig. 4. UMG—Gradient formation latencies (WC = Worst Case scenario see Eq. (4)) (BC = Best Case scenario see Eq. (6)).

messages to achieve such functionality which, in WSN, results in a expensive resource consuming activity that cannot be afforded. The idea of introducing a service advertisement and discovery mechanism at the routing layer has been implemented in UMG. The process of service advertisement/discovery is implemented as part of the routing process by making use of every SpreadGrad message traversing the network. When creating the gradient, the SpreadGrad message carries a descriptor of the service/data provided by the gradient origin node (see "descriptorBF" in Fig. 2). The metadata set, M, is a collection of human readable words in a dictionary which is common to the whole network. Items in M are employed to define the type of service or the behaviour of a node in terms of data acquisition, i.e. interest. The "descriptorBF" structure contains a subset of the metadata, D, such that $D \subseteq M$, and D defines the service of a gradient origin node. The idea lays in the possibility of extending the universe of M while not affecting the "descriptorBF" of the nodes in the network. To achieve this, the "descriptorBF" field operates as a Bloom filter [29], i.e. an array of bits set to 1 and 0, which store the membership of an item in the universe of M, rather than storing the item itself. Due to the fact that storing every item (word) would cost too many bytes, and associating one bit of a bitvector to each item restricts the scope of M, Bloom filters were selected as a compressing structure to store and advertise the metadata/services.

On reception of a SpreadGrad message, a node updates its routing table with the "descriptorBF" for its gradient origin node entry (see Fig. 3). This mechanism allows for each node to have a descriptive map of the type of services offered by those nodes for which the local node is a router, along with the distance in hops to reach them. This is a powerful tool for the decision making process of applications working on top of the routing protocol in areas like distributed data fusion, discovery service, distributed storage or swarm decision making. Moreover, the Bloom filter descriptor can be seen as a profile descriptor which describes the node and thus could serve as a mechanism to create overlay virtual communities of sensors. UMG provides an interface to the upper layers to create and query Bloom filters. An application might contain a set of keywords, i.e. a dictionary, which defines the data offered/contained/stored by the node. With this approach, an implicit ubiquitous lookup system is available which provides hints in the searching process.

In order to populate the "descriptorBF" structure before spreading the gradient, a set of keywords from M are selected which describes the behaviour of the node. A set of hash functions are applied to each keyword to produce a set of positions in the array of bits of the Bloom filter which are to be set to 1. Once created, Bloom filter structures are easy and fast to query and can be compared using a "XOR" bitwise operation. However, the Bloom filter structure has a drawback. The higher the number of data items hashed in the Bloom filter, the higher the probability of obtaining false positives. A false positive occurs when a data item has not been hashed in the Bloom filter, but the bit positions which correspond to the data item are set to 1 as a combination of positions from other data items previously hashed. The selection of the number of hash functions, together with the number of bits of the Bloom filter and the number of elements inserted in the Bloom filter, establish the probability of getting false positives. On the other hand, the Bloom filter guarantees that there will be no false negatives, i.e. if there is not a combination of positions set to 1 which satisfies the hashed data item, then it is guaranteed that the data item was not inserted in the Bloom filter.

A function to compare the similarity of two Bloom filter structures, according to a percentage value of accuracy, is also implemented. The accuracy percentage is obtained as the fraction of the number of matching positions set to 1 in both Bloom filters divided by the number of positions set to 1 in the Bloom filter with the highest number of positions set to 1. This functionality is employed by UMG to query the routing table of other nodes descriptors for the discovery of services. Moreover, this mechanism provides support for applications which require searching or aggregation of different types of data. In UMG, the default parameters employed to configure the descriptor Bloom filters are: size equals to 32 bits, i.e. DESCRIPTOR_SIZE, and number of hash functions equals to 5. If the application contains a set of predefined keywords M, i.e. a metadata dictionary, the Bloom filter can be firstly tested to reduce the probability of obtaining false negatives on selected subsets.

3.4. Data transport phases

This section describes the design and implementation of the three phases employed in the transportation of the data. These phases are launched when the gradient for the destination, and even the source, node has been spread. The three data transport phases are: (1) Gradient Descent Phase, (2) Local Repair Phase, and (3) End-to-End Acknowledgement Phase.

3.4.1. Gradient descent phase

This phase is executed when a node needs to communicate with another node which has already spread its gradient. The Routing Table is consulted for the entry corresponding to the gradient origin node address. The Routing Table can also be searched by descriptors, "descriptorBF", searching for nodes which can satisfy the requested service (see Fig. 3). If the entry in the Routing Table cannot be found and the local node requires communication with a specific node, then the local node needs to setup its gradient with the option of requesting a particular address, or a set of nodes matching the "requestDescriptorBF", such that the requested node/s spread the gradient. When there is a Routing Table entry for the destination node which is marked as enabled, the local node prepares the "DataGrad" message (see Fig. 5). Data from higher layers is encapsulated in the "data" field to be transported. A new sequence value is generated by the originator node as "seq" in the "DataGrad" message. Together with the originator node address encapsulated in the "originAddr" field and the destination node address as the "originGradAddr", the packet is uniquely identifiable. The packet is sent to the address indicated by the "receivedNodeAddr" field in the Routing Table. Intermediate nodes receiving the "DataGrad" message keep on forwarding the data packet to the address of the node in

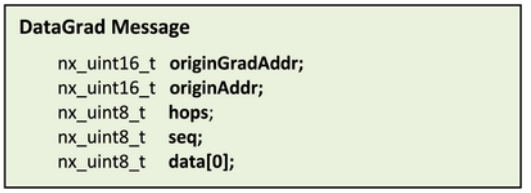

Fig. 5. UMG—DataGrad message structure.

the "receivedNodeAddr" field of their Routing Tables for the corresponding gradient origin node entry, i.e. the "originGradAddr" field. The number of hops is increased in the "DataGrad" message ("hops" field) with every new node visited. The "hops" field is also used as an incremental hop counter when climbing the gradient towards the originator of the communication in the End-to-End Acknowledgement Phase (see Section 3.4.3). This way, the "hops" field is employed to update the "realHops" field of the Routing Table in the intermediate nodes, which indicates the most recent real hop distance.

Achieving Reliability when Descending the Gradient:

The wireless medium is prone to errors in the packet transmission due to collisions, noise, or environmental condition changes which can last for an arbitrary length of time. Communication between neighbours might not be established for short periods of time due to congestion at nodes, the node being in a sleeping mode, or the medium being contended. The connection can also be blocked for a long period of time, for example an obstacle could be temporarily interrupting the communication or local mobility might be placing nodes too far apart for the link to be stable. Moreover, a link might be broken as the next node might have died or changed position. In any of these situations, reliability mechanisms are employed to achieve end-to-end communication in the path.

In this regard, UMG employs acknowledgement packets, timers, and retrials for the end-to-end reliability of the path. If an explicit end-to-end acknowledgement packet is not received by the originator of the communication after a predefined time, the packet is sent again. This procedure is repeated for a number of trials until the packet is acknowledged. The number of trials is subject to be changed and by default has been set to 2. When the maximum number of trials is reached, the option of employing the gradient creation phase, with or without the option of requesting for the destination node to spread its gradient, can be launched to create routes. To reduce the number of gradient formations, which are expensive in terms of communication, UMG enables the acknowledgement mode at the MAC layer. In this mode, a unicast packet issued by a node must be explicitly acknowledged by the next hop receiver. This mechanism allows UMG to implement a procedure for which a node waits for a period of time for an acknowledgement packet and resends the packet up to a maximum number of trials if the acknowledgement packet is not received. In addition, UMG employs snooping functionality to eavesdrop the data packet being sent by the next hop node as a backup mechanism to implicitly acknowledge the packet. Both mechanisms increase the reliability of the data packet delivery in situations where short time communication problems occur. Additionally, when the communication between two neighbour nodes repeatedly fails, the maximum limit of retrials will be reached and no acknowledgement will be received by the next node when descending the gradient. In this situation a local repair mechanism is started which finds the most appropriate neighbour node to keep on descending the gradient.

Two reliability modes can be configured in UMG which provide two alternatives to enforce reliability depending on the type of application and traffic conditions. Each data packet can be configured at the originator node with two Reliability Modes: (1) Requesting End-to-End Acknowledgement or (2) Disabled End-to-End Acknowledgement. In Reliability Mode 1 the final destination sends an end-to-end ack packet back to the originator node (see Section 3.4.3), while in Reliability Mode 2 end-to-end ack packets are not issued by the destination on reception of a data packet. The Reliability Mode of each data packet is controlled with the "hops" field (see Fig. 5); Mode 1 starts in 0 while Mode 2 starts in 127 (subtracting 127 at each hop to calculate the number of hops) when the originator nodes sends the packet. The 255 values of the 1 byte reserved for the "hops" field (note that 0xFF is also reserved) can be comfortably split into two segments to identify the two modes of the data packet as it is not feasible to route data over long routing paths in wireless sensor networks. While Mode 1 produces an explicit confirmation at the originator of the arrival of the packet to the destination, Mode 2 provides a reliable alternative to rapidly push data in one direction and to reduce network traffic without notifications to the originator node. The use of UMG without end-to-end acknowledgements has resulted in better and more efficient performance in terms of delivery ratio, latency and overhead communication costs, especially in scenarios with high degree of noise and traffic. However, both modes can be dynamically interchanged in a per packet manner according to the necessity of the application at each time.

3.4.2. Local repair phase

The local repair mechanism is launched when (i) the local node needs to send a message to a node for which there is no entry in the Routing Table or (ii) the maximum number of trials is reached and still the data packet has not been acknowledged by the next neighbour in the gradient descending path. In this phase, the node waiting to receive the acknowledgement packet needs to look for an alternative neighbour node which can keep on relaying the packet towards the gradient origin node. For this purpose, a neighbourhood discovery process is launched by broadcasting a "Neighbours" message (see Fig. 6). The message is typed as

```
Neighbours Message
    nx_uint8_t   type;
    nx_uint16_t  originGradAddr;
    nx_uint8_t   descriptorBF[DESCRIPTOR_SIZE];
    nx_uint16_t  brokenAddr;
    nx_uint8_t   hops;
    nx_uint8_t   seqGradient;
```

Fig. 6. UMG—Neighbours message structure.

LOCAL_REPAIR and contains the address of the node which has failed to acknowledge the packet ("brokenAddr"), and the distance in number of hops ("hops") from the sender node to the gradient origin node ("originGradAddr") (see Fig. 6). The message also contains the service descriptor ("descriptorBF") of the sender of the message which serves to provide extra information to neighbour nodes; this can act as a filter to enhance or restrict their reply (see Section 3.5.1). The "brokenAddr" and "hops" values are included to restrict the replies to those neighbours which can provide a valid route towards the "originGradAddr" node. A candidate node must not contain the address of the sender of the "Neighbours" message, nor the value of the "brokenAddr", in the "receivedFromAddr" field of the "originGradAddr" entry in its Routing Table. In other words, the next node to reach the "originGradAddr" in the candidate node's Routing Table cannot be the node which failed to acknowledge the packet, nor the sender itself. However, if there are no neighbour nodes replying which conform to these restrictions, the next trial of the local repair includes the node which previously failed, i.e. "brokenAddr", as it may have suffer a short temporal disconnection. In addition, the candidate node must be at least equally close, in terms of hops, to the "originGradAddr" node than the sender node is. These checking procedures avoid the formation of cycles which might incur a higher number of local repair mechanisms being launched, increasing the contention in the medium and decreasing the performance.

However, when gradient updates for the same gradient origin node occur, it is possible that some nodes will not get updated, and thereby contain the address of the next hop ("receivedFromAddr" field) in their Routing Tables corresponding to older gradient sequences "seq". In this situation, multiple gradient updates, with different sequences, need to coexist such that loops are not formed when performing local repairs. In Fig. 7, this effect can be seen where a new gradient setup has been received which only updates some of the nodes in the network while the rest of the nodes still contain old gradient entries in their Routing Tables. In the diagram, for instance, node 10 issues a packet to descend the gradient towards node 1 via the new gradient. When descending the gradient, node 6 relays the packet to node 5 which does not acknowledge the packet after a number of trials. Thus, node 6, which is at 4 hops distance to node 1 via the new gradient, starts a local repair phase to discovery a valid next hop node. The only reply to the LOCAL_REPAIR packet sent by node 6 comes from node 8, which is also at 4 hops distance from node 1 but under the old gradient setup. Node 8 is a valid next hop node, as its next hop address is node 7 which does not pose any restriction. Nevertheless, node 7 relays packets to node 6 under the old gradient setup; this creates a loop. In order to avoid loops in this situation, UMG restricts a neighbour node from being a valid candidate if the sequence value "seq" of the entry "originGradAddr" in its Routing Table is older than that of the sequence of the node starting the local repair phase. The sequence of the initiator of the local repair phase is transported in the "Neighbours" message in the field "seqGradient" when the

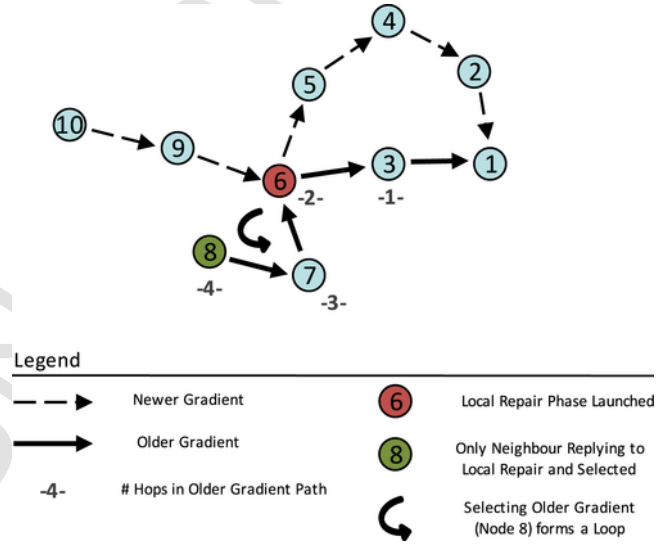

Fig. 7. UMG—Local Repair Phase—risk of loop formation when coexisting multiple gradient updates.

packet is typed as LOCAL_REPAIR. Nodes receiving the LOCAL_REPAIR message reply if their sequence for the "originGradAddr" is newer than the sequence received in the packet ("seqGradient"). Since sequence values roll over when reaching the maximum value of $2^{?8}$, a way to identify the new sequence when compared to others is required. For this purpose, the sequence counter is considered as a cycled structure where the newest sequence value is that with the highest distance to the other sequence value being compared against—the rationale behind this is that the probability of a small number of gradient setups updating all the nodes in the scope is higher than the probability of a node not getting updated when a larger number of gradient setups occur. Loops are not formed when packets progress from a node belonging to an old gradient setup to a node involved in a newer gradient formation. Thus, when selecting a node to repair a gradient, newer sequences will overlap older sequences even if the hop distance is higher. In addition to this, and as a precaution measure against man-in-the-middle attacks, UMG also implements a mechanism to detect cycles (see Section 3.4.3).

After a LOCAL_REPAIR message is sent, valid candidate neighbours reply with a "Neighbours" message with the "type" field set to REPLY_LOCAL_REPAIR. The latter message contains the number of hops to reach the "originGradAddr" node. From the list of candidates, nodes with the newest sequences are considered. From them, the closer one, in terms of hops to the gradient origin node, is selected. Other metrics like signal strength or link quality could enhance the decision. However, UMG randomly selects a node from those closer to the "originGradAddr"—this process is efficiency in terms of complexity, processing and memory. The gradient descending process is resumed employing the selected candidate node as the next hop neighbour. The Routing Table is updated accordingly, with the candidate neighbour as the next neighbour to reach the "originGradAddr".

If a candidate neighbour is not found, i.e. the local repair fails, two procedures can occur depending on the Reliability Mode (see Section 3.4.1) of the data packet:

1. In Reliability Mode 1 (end-to-end acks are enabled) when the local repair fails, instead of doing a backtracking process, the intermediate node disables the Routing Table entry for the "originGradAddr" and discards the packet. When the end-to-end timer expires at the originator of the communication, and the maximum number of trials is not reached, the data packet descends the gradient again. In this case, a local repair process will be launched just one hop before the intermediate node which could not repair the gradient previously. This can be seen as an induced backtracking process which is always initiated by the originator of the communication. In this way, if the maximum number of end-to-end trials is reached, UMG opts to update its gradient since the gradient is experiencing a high degree of instability and failure.
2. In Reliability Mode 2 (end-to-end acks are disabled) when the local repair fails, the intermediate node triggers a gradient formation process requesting the gradient origin node, i.e. the destination of the data packet, to spread its gradient. Consequently, the gradient will be repaired for both the intermediate and the originator node of the data packet which will be sent by the intermediate node towards the new gradient.

3.4.3. End-to-End acknowledgement phase: short memory cache

The End-to-End Acknowledgement Phase, if enabled, starts when the destination node is reached, i.e. the "originGradAddr". The destination node provides the data to the higher layer and is in charge of issuing an end-to-end acknowledgement packet back to the originator of the end-to-end communication, i.e. the "originAddr" field in the "DataGrad" message. The end-to-end acknowledgement packet is a "DataGrad" message with no data enclosed (see Fig. 5). Rather than having a dedicated field for the packet type, the end-to-end acknowledgement packet is identified by its length as it only contains the header fields of the routing protocol. The end-to-end acknowledgement packet is configured with the same values of the data packet being acknowledged. The "hops" field is reset to 0 and is increased at each intermediate node when climbing the gradient, in order to update the "realHops" field of the Routing Table (see Fig. 3). It has to be noted that the "lastTimeUsed" field of the Routing Table is also updated. Both the "realHops" and the "lastTimeUsed" fields provide an indication on the status and distance of the end-to-end communication, useful for decision making in higher layers.

The end-to-end acknowledgement phase depends on a caching mechanism which stores information about received and sent data messages when descending the gradient. Every time a data packet is received, the key values defining a unique message are stored in the caching structure. The structure caches the key fields of the "DataGrad" message in a small array. When a packet arrives, this is stored as "status" equals to "MEM_RECEIVED" and the "timeStamp" is set to the actual time of the mote. This mechanism is used to control cycles and avoid the reception of duplicate packets. When the node sends the message to the next node in the gradient, the entry for the message changes its "status" to "MEM_SENT". This mechanism differentiates received packets in the "CacheMemory" from those which are also waiting to be acknowledged by a packet coming from the final destination (gradient origin node).

When the end-to-end acknowledgement packet is broadcast on the way back to the source node, neighbour nodes receiving the packet check their "CacheMemory" array for a match. If the packet is found and its status is set to "MEM_SENT", the node increases by one the number of hops in the end-to-end acknowledgement packet ("hops" field), updates the "realHops" field in the Routing Table, and broadcast the end-to-end acknowledgement packet. The only nodes broadcasting the end-to-end acknowledgement packet are those which have previously forwarded the data packet when descending the gradient.

Furthermore, the link might fail due to a series of short and long time communication problems (see Section 3.4.1). Therefore, a reliable mechanism is in place to deliver the end-to-end acknowledgement packets in each gradient ascending hop. Due to the fact that broadcast packets are not acknowledged at the MAC layer, the UMG routing protocol snoops packets searching for

a broadcast packet with a higher value in the "hops" field. If the packet is snooped within a certain time, the entry in the "CacheMemory" array is disabled, i.e. "status" changes to "MEM_DISABLED". On the contrary, if the timer expires, the end-to-end acknowledgement packet is broadcast again. This occurs a predetermined maximum number of times. If after that, the message could not be acknowledged, i.e. snooped, it is assumed that bidirectionality in the path cannot be achieved and the packet is discarded. However, the gradient spreads in the direction of the climbing process and the likelihood of establishing communication in the ascending direction should be higher than when descending the gradient. In the situation where a packet is discarded, the end-to-end timer at the source node will fire; the node either sends the data packet again or spreads its gradient while requesting the destination node to do the same. Additionally, UMG offers the possibility to utilize the gradient of the originator of the communication to keep on forwarding the end-to-end acknowledgement packet, providing there is an entry in the Routing Table for the gradient origin node. If there is an entry, the procedure is the same as in Section 3.4.1, including the local repair mechanism in Section 3.4.2. For this particular purpose, the "hops" value in the end-to-end acknowledgement packet is set to its maximum, thereby acting as a flag and missing the counting functionality ("realHops"). This flag indicates to the UMG protocol that the packet is an end-to-end acknowledgement message thus memory caching is not performed and the end-to-end acknowledgement at the destination node will not issue an end-to-end acknowledgement packet on reply. Instead, when the originator of the communication receives a data packet with this flag, it will be treated as an end-to-end acknowledgement, stopping the end-to-end timer. When bidirectionality does not exist and there is no gradient setup for the originator node, the end-to-end communication would not be achieved. In this situation the originator node spreads its gradient by querying the destination node, both gradients will be setup and end-to-end communication will occur one way or another.

3.5. Mobility support in wind turbine structures

The UMG routing protocol has been designed to support dynamically changing topologies, where gradient origin nodes leave their current neighbourhoods to become part of other areas. UMG tends to reduce radio activity when there is no data communication in the neighbourhood, avoiding the use of periodic hello messages. The premise is that the network updates itself via the use of opportunistic communication, either received or snooped. Areas with frequent data communication activity will make use of overheard packets to detect when a node is changing its position with respect to its neighbours, i.e. relative mobility. On the other hand, areas with no communication activity will not be updated even if mobility occurs. These areas, where communication occurs infrequently, might require on-demand gradient updates if the topology changes. However, they will require low maintenance in terms of communication overhead and route maintenance.

The UMG routing protocol integrates a component to estimate whether a node has changed neighbourhood, which employs a probabilistic mobility assessment mechanism described in previous works [52]. The mechanism, which reports to UMG, operates as a single component and is fed with all packets received or snooped. The address of the sender of each of the overheard packets is cached in memory efficient structures, Bloom filters. By employing a structure of temporal shifting Bloom filters, each filter stores the address of the neighbour nodes for a period of time before the next Bloom filter starts to be filled out. Periodically, a probabilistic estimation mechanism, employing a predefined table model, utilizes the information stored in the Bloom filters to determine whether the node's neighbourhood is changing. This process compares previous states of temporal Bloom filters against a master Bloom filter. The master Bloom filter contains information of the core neighbourhood obtained with a deterministic query process in a past period of time. This way, previous states of mobility can be recursively computed to decide on the probability of the node changing neighbourhood. When mobility is detected, UMG employs explicit local update messages to confirm the neighbourhood mobility status, to discover the number of new and old neighbours, and to populate its master Bloom filter. All the entries in the routing table are disabled as they will be of no routing use in the new neighbourhood, but they are not deleted as they are employed for service lookup. In this case, the node will update its routing table opportunistically based on the communication being received. If the node is a gradient origin node, it needs to create routes from its old neighbourhood to its new position. To do so, the gradient is spread with a limited scope in terms of hops, depending on how far the node is from the farthest old neighbour. This mechanism reconfigures only nodes within the area of the scope such that they forward data packets towards the new neighbourhood. An expensive global flooding will not be necessary as the scope of the flooding only has to reach the old neighbours for the gradients to keep converging towards the node.

The above mobility assessment mechanism has been proven to be effective in detecting, and alerting the UMG routing protocol of, mobile nodes, predominantly in networks where the majority of devices are static and transient nodes traverse an area at varying speeds or arrive to an area to become static nodes. However, wind turbine monitoring presents a new scenario where nodes installed at the blades exhibit semi-deterministic mobility patterns when blades are spinning. This behaviour produces cyclic intermittent connections between transiting nodes and static nodes dependent on location. While opportunistic routing could be one solution, the speed of transient nodes, the link quality and the environmental conditions can have a negative impact on the performance of the routing process. Thus, a different approach, which regulates how routing is performed from, to, and through these type of mobile nodes, is required. For this purpose, UMG has been equipped with a mechanism to control the routing's gradient formation process known as Inter-Cluster Gradient Formation (ICGF).

3.5.1. Inter-cluster gradient formation

The Inter-Cluster Gradient Formation (ICGF) mechanism operates when gradients are created or updated selecting which of the received gradient formation packets are employed to populate the routing protocol table depending on the sender of the packet. In other words, the ICGF mechanism enables the user to leverage information from the supervised deployment of the nodes to improve the gradient formation process. The ICGF mechanism provides a simple interface to the application layer for the formation of clusters of nodes which belong to special areas in the network. Each cluster is given a tag, such as "tower cluster", "blade cluster" or "nacelle cluster", and nodes are assigned membership on a cluster. In addition, the interface enables the user application to define the so-called inter-cluster gradient rules. These rules define which clusters can interact to form a gradient and which are not allowed, despite some of their nodes being in radio communication range.

The ICGF mechanism operates as follows. When the Gradient Spread Phase takes place for any node (see Section 3.2), every node sending a SpreadGrad message will make use of its service advertisement mechanism (see Section 3.3) and encapsulate its cluster tag in the descriptor field of the message (see "descriptorBF" in Fig. 2). This way, nodes receiving the SpreadGrad message check whether the descriptor of the message contains one of the cluster tags to which the cluster of the local node is allowed to interact with according to the inter-cluster gradient rules. Thus, the ICGF mechanism serves to filter the type of gradient formation packets received before their information is used to update the gradient in the routing table. For instance, a node belonging to a cluster tagged as "blade 1" could only accept gradient messages from nodes belonging to a cluster tagged "nacelle" while not considering messages received from nodes belonging to a cluster tagged "tower". This way, the ICGF mechanism enforces the creation, or update, of gradients between the tower and blade 1 through a node placed at the nacelle. Consequently the gradient route will be more reliable as the route is more stable and does not employ transient nodes as routers. It has to be noted that the ICGF mechanism can also perform in networks where the deployment of only a set of partial nodes is supervised. However in certain scenarios where the connectivity of the network might be low, the ICGF mechanism can impact on the formation of gradients as it would limit the interaction between neighbours. For these situations, when ICGF is enabled and is limiting the proper formation of gradient paths, the application layer can request the ICGF mechanism to spread the gradient with the tag "ICGF Disable" in the descriptor. This tag, which will be checked first when a packet is received, disables the ICGF filtering process. The ICGF mechanism is also employed in the local repair phase (see Section 3.4.2) when the gradient is being repaired. The "descriptorBF" in the "Neighbours" message (see Fig. 6) is also used by the node attempting to repair the gradient for restricting potential candidates neighbours in their reply according to the cluster tag and the inter-cluster gradient rules.

3.6. Dynamic sleeping mechanism

In order to prolong the lifetime of battery-powered nodes, and thus preserve the utility of the whole network, it is paramount to optimize energy usage when possible. The most energy-draining activity for a sensor node is radio communication, and a significant portion of that cost can be attributed to idle listening. Methods to decrease this overhead fall into the area of Low-Power Listening (LPL). There has been a wide range of LPL methodologies presented in the literature which aim to minimize idle listening time, an energy-demanding radio state. These methodologies configure the proportion of time the radio spends in the listening/transmitting state, i.e. the duty cycle of the radio transceiver. More precisely, Clear Channel Assessments (CCA) are performed by the radio at regular intervals in order to sense the channel for a carrier. If a carrier is detected, the MAC layer prepares to receive a packet. Otherwise, the controller returns to sleep for a period of time, i.e. the sleep interval, which determines the duty cycle. However the duty cycle applied to the transceiver of nodes needs to be carefully controlled when routing packets in the network. The main goal seeks to maximize the time the radio is sleeping while at the same time provide high quality service in routing when traffic is ongoing in the neighbourhood.

A popular approach in LPL consists in defining only the sleeping time. After this time, the transceiver wakes up and listens for activity in the medium. If there is no neighbourhood radio activity for a very small period of time, enough to discover a preamble signal, the transceiver goes to sleep again for the period defined by the sleeping time. If radio activity occurs, or the local node needs to send a packet, the transceiver is awake until activity ceases. An adaptive solution which dynamically regulates the sleeping and active times according to the type of traffic and routing protocols is desired and is work in progress. However for the purpose of testing and comparing the performance of the UMG routing protocol operating in LPL mode, a simple dynamic sleeping mechanism has been implemented. The LPL mechanism operates similarly to the previously described one with the exception that the active time is defined initially and establishes a minimum time for which the transceiver is active after a sleeping period. If the active time is elapsed and communication in the neighbourhood is still ongoing, then the active time is extended by an extra period of time from the last time communication occurs. The extra period of time is added to the active time to maintain the radio active when there is activity in the neighbourhood. The transceiver goes back to sleep only when there is no communication in the neighbourhood for the duration of the last extra period of time. The extra period of time is defined according to the maximum delay in the retransmission of any kind of packet by the UMG routing protocol. In this case, it has been set to double the delay time in between broadcasts when setting up the gradient to guarantee that a node remains awake while the gradient is being spread in the area. The minimum value for the transceiver active time has been set equal to that of the extra period of time. Nevertheless, it has to be noted that there will be a tradeoff when setting the sleeping time; long sleeping times will have an im-

pact on the proper formation of gradients and in the latency of data communication while short sleeping times will drain energy quicker at the benefit of potentially improving the performance of the routing process.

4. Evaluation of UMG

This section studies the performance of the UMG routing protocol in a wireless sensor network operating on a wind turbine structure. UMG is also compared with the de-facto routing protocol for collection in sensor networks, i.e. the Collection Tree Protocol (CTP) (see Section 2.1.1). The evaluation environment is initially described as well as the three enhancements to the latest version of the simulator, TOSSIM 2.x, which have been made to enable (i) node mobility, (ii) radio duty cycle, and (iii) energy consumption tracking (PowerTOSSIM-Z). Next, the experimental procedure and settings are explained. The network topology for a wind turbine structure is presented along with the variety of scenarios for testing. Thereafter, UMG and CTP are evaluated and their results analysed and compared under various connectivity, traffic and radio duty cycle conditions. UMG outperforms CTP in the diverse scenarios tested and is presented as an efficient and effective solution for wireless sensor routing in this type of dynamic structures. Finally, UMG's overall performance is analysed and some guidelines for the best operating conditions of UMG are deduced from the results.

4.1. Evaluation environment: TinyOS and TOSSIM

The protocols have been implemented and evaluated in TinyOS v.2.1.1 [38], one of the most commonly used operating system for WSN, through the use of TOSSIM [53–55], an open source system simulation environment included in the TinyOS suite. TOSSIM can capture a wide range of network interactions, and is capable of scaling up to over one thousand nodes by simulating using a fine-grained, bit-level granularity [53]. TOSSIM is a discrete event simulator, such that the operation of the system is driven by the execution of chronologically ordered events which react to, and change, the state of the system [56]. TOSSIM is integrated in the TinyOS suite as a library which enables the user to control and configure the simulation environment (such as topology, noise floor, radio and MAC models). It makes use of most of the code in the stack of a TinyOS application to generate the executable and replaces low level chip components with an implementation of their behaviour. Currently, TOSSIM supports simulation of the main chips integrated in the micaZ sensor platform (see Section 4.1.1), the Atmel ATmega 128L microprocessor [57] and the Texas Instruments CC2420 Chipcon Transceiver [58]. TOSSIM maps directly to the TinyOS code which reduces the gap between the simulator and the real environment. By replacing low level components, a high level of fidelity between the functionality of the system in the simulator and in the real device is achieved. This also enables testing the codebase of an application both at node and network levels before testbed deployment.

The MAC object in TOSSIM controls variables such as backoff, packet preamble length, radio bandwidth, etc., and by default is configured according to the CC2420 transceiver chip operation. In addition, the radio propagation model in TOSSIM is based on the propagation strength of each node and the noise floor trace introduced by the user. Other parameters such as the receiver sensitivity are configured according to the CC2420 transceiver specification. The radio propagation model employs a Signal to Noise Ratio (SNR) curve derived from experiments with two micaZ motes. Additionally, noise floor and interference from the environment and other nodes are included in the model via the use of the Closest-fit Pattern Matching (CPM) algorithm. As described by Lee, Cerpa, and Levis in [55], "the CPM model significantly increases the accuracy of the simulation in terms of packet delivery by acknowledging the time-dependence of wireless noise". "This model can capture bursts of interference and other correlated phenomena". The CPM model exploits the non-linear behaviour of the relation between the packet reception ratio (PRR) and the signal to noise ratio (SNR). The user is responsible to create the topology in a connectivity file where unidirectional gain values between sender and receiver nodes in the network are assigned. Each gain value indicates the signal strength in dBm at which the destination receives the signal. Additionally, CPM generates a statistical model from the noise floor trace file selected by the user. Two noise floor files included in TOSSIM have been considered which have been obtained from different environments. The "Casino-lab" noise floor trace file has been obtained from measurement at a laboratory in the Colorado Schools of Mines, which produces a lower level of noise with an average dBm value of −97.69 and standard deviation of 1.34 calculated from a subset of 5000 samples from the noise trace file. The "Meyer-heavy" noise floor trace file corresponds to the very noisy environment in the Meyer library from Stanford with an average dBm value of −93.21 and standard deviation of 8. These files are employed to test and compare networking protocols operating on a specific topology under different levels of attenuation. Depending on the path attenuation between two nodes (indicated in the topology file), the SNR can be calculated from the noise floor, and according to the SNR/PRR curve, the packet drop rate is obtained.

4.1.1. The MicaZ wireless sensor platform

The micaZ [59] wireless sensor device (see Fig. 8), originally developed by the University of California, Berkeley, can be programmed with the TinyOS open-source operating system and is employed as the reference sensor platform for the implementation of the low level components in the TOSSIM simulation.

The micaZ platform incorporates an 8-bit Atmel Atmega128L micro controller [57] with an 8 MHz CPU. It integrates the following memory modules: 4 kB SRAM, 128 kB flash memory for program code, and 512 kB flash memory for measurement/data storage. The current consumption of the micro controller in Active mode is 8 mA, while less than 15μA in Sleep mode

Fig. 8. MicaZ wireless sensor platform.

[59]. The micaZ mote integrates the 2.4 GHz Texas Instruments Chipcon CC2420 [58], which is IEEE 802.15.4 [60] compliant. The CC2420 transceiver transmits in the ISM 2.4 GHz band (2400–2483.5 MHz) with a transmission data rate of 250 kbps. The IEEE 802.15.4 specifies 16 channels within the 2.4 GHz band, in 5 MHz steps, numbered 11 through 26. The device integrates an antenna capable of reaching up to 100 m outdoors and 30 m indoors. The output power register in the CC2420 transceiver varies from level 3 at −25 dBm and a consumption of 8.5 mA, to level 31 at 0 dBm with a consumption of 17.4 mA. Accordingly the transmission range also varies which also depends on the environment where the mote is placed. On the other hand, when the transceiver is in receiving mode, the current drawn is 23 mA. Therefore, it can be observed that the most expensive activity in terms of power in the transceiver, and also in the whole architecture of the device, is the reception of a message (RX mode). The reception mode needs to be active not only for receiving a packet but also while listening. This is the main reason why efforts to save energy in wireless sensor devices focus on the design of low power policies which duty cycle the transceiver from RX mode to Sleep mode (with a current consumption of 1μA). The micaZ is powered by two AA batteries and it comes with a 51-pin UART expansion connector for communicating with expansion and gateway interfacing boards, e.g. the MIB520CB board [61] employed for programming and gateway connectivity purposes. The micaZ sensing board interfaces through the 51-pin expansion connector to provide add-on sensing capabilities. For instance, the MTS310 board (see Fig. 9) is a flexible sensor board including a Dual-Axis Accelerometer, Dual-Axis Magnetometer, Light, Temperature, Acoustic and Sound.

4.1.2. PowerTOSSIM-Z: realistic energy modelling in TOSSIM

For the purpose of simulating realistic power consumption in TOSSIM version 2.x, an energy consumption plug-in was developed. The new plug-in, called PowerTOSSIM-Z [62], is an upgrade from PowerTOSSIM [63] for the purpose of tracing power consumption in wireless sensor networks. In this work, the existing PowerTOSSIM plug-in functionality was upgraded to version 2.x of TinyOS, while support for the micaZ hardware that incorporates a realistic, non-linear energy model was included. Tracing the operation of different components connected to the CPU (leds, spi, …) was also included. External peripherals like flash components were also upgraded, supporting erasure and crc operations. The micaZ energy model was defined and extra functionality was introduced in the radio model component to support different transmission power levels. The in-line execution of these components was traced with debug statements and realistic time state transitions were considered. The post-processing of the energy trace and its evaluation against a realistic battery model which was designed taking into account that voltage drops monotonically in a non-linear manner as a function of the capacity remaining and the chemistry of the battery. The model also takes into account the two main effects in the behaviour of batteries: (i) the rate capacity effect, which defines that higher discharge rates lead to a lower residual capacity and (ii) the recovery capacity effect, by which batteries in idle mode can recover some charge.

Fig. 9. MTS310 sensor board.

4.1.3. Low-power listening in TOSSIM

TinyOS provides a low-power listening layer for the CC2420 transceiver, the radio module employed in TOSSIM. The low-power layer controls the duty cycle period: it switches the state of the transceiver from reception to idle and vice versa according to low-power policies, and turns it on when a packet is to be transmitted. The transceiver is put to sleep or awoken for the reception of packets according to both the duty cycle strategy and the clear channel assessment (CCA). However, there is no low power layer support for TOSSIM. In this regard, an adaptation of the CC2420 low power layer has been implemented which allows control of the status of the transceiver in the simulator. The implementation, which does not consider CCA per se due to TOSSIM being a discrete event simulator, serves as a mechanism to duty cycle the transceiver. It takes into account when a packet is being received and enables the transceiver when a packet is to be transmitted. The sleeping and active times can be defined and the implementation follows the dynamic sleeping mechanism in Section 3.6. Through the use of the PowerTOSSIM-Z plug-in, the energy consumption can be evaluated for different duty cycles.

4.1.4. Simulating mobility in TOSSIM

The most recent version of TOSSIM, version 2.x, does not enable node mobility. For the purpose of simulating mobility in TOSSIM version 2.x, the core of the simulator has been enhanced in order to accept topologies where nodes change position with respect to their neighbours. The implementation (see [64]) allows simulation designers to change the gain (dBm) on any link between two nodes at a specified time in TOSSIM. A tool to generate the correlated set of static topologies which represent the connectivity variation amongst transient nodes in the network has been created. The tool makes use of the link layer model generator from Zuniga and Krishnamachari [65] (distributed with TOSSIM) to generate the gain-based static topologies required in TOSSIM. The implementation has been tested and calibrated against the ns-2 simulator [66], taking into consideration that TOSSIM exhibits a higher level of real-world behaviour. The mobility tool has been employed to create topologies for TOSSIM which follow the structure and behaviour of wind turbines. Nodes placed at the blades have been rotated at different speeds according to position and dimensions of the wind turbine structure.

4.2. Experimental procedure

For evaluation purposes, a series of settings and metrics have been employed to assess the performance of the protocols under a wide range of test case scenarios. The methodology to analyse the performance of the routing protocols is explained.

4.2.1. Experimental settings

One of the largest wind turbine structures, the Siemens D6 platform SWT-6.0–154 [67], has been selected to simulate the operation of a wireless sensor network deployed in these structures. The Siemens 6 MW offshore wind turbine has three 75-m long blades amounting a 154-m rotor diameter which sweeps an area of $18\,600\text{m}^2$ to maximize energy yield at offshore locations (see Fig. 10). The rotor speed ranges from 5 to 11 rpm and the tower height can be adjusted according to the specifications of the site.

A topology has been created to simulate a realistic wireless sensor network deployment following the dimensions and rotor speed range of the Siemens SWT-6.0–154 (see Fig. 11). Nodes 0–4 have been placed along the tower and nodes 5–10 at the nacelle to simulate monitoring of different components and structural elements. Nodes 11–13 have been installed on blade 1, nodes 14–16 on blade 2 and nodes 17–19 on blade 3. Nodes on the blades and the tower are distributed with an inter-node distance of 25 m. The set of nodes placed along the blades are expected to rotate clock-wise at 6 and 10 rpm during the simulation and evaluation in accordance with the specifications of the Siemens SWT-6.0–154. The TOSSIM mobility plug-in has been configured to update the position of the sensors on the blades every second according to the selected rotational speed.

The transmission range of the nodes has been configured to 37 m. This has been achieved using the link layer model generator from Zuniga and Krishnamachari [65] which produces a TOSSIM compliant gain-based topology based on the geographical coordinates of nodes in the network. The tool needs to be configured with the parameters involved in the log-normal shadowing path loss propagation model which establishes the attenuation factor as a function of the inter-node distance. Through experiments carried out in TOSSIM, using the "Casino-lab" noise floor trace, the transmission range corresponding to the variation of parameters in the tool has been calculated when packets cease to be received by a node moving away from the sender. In addition, the transmission range has been averaged from the set of repetitions of the same experiment.

The noise floor in the network has been varied in the experiments. The two noise floor traces selected differ from each other in the average noise floor value and the standard deviation of the sample distribution which impacts the packet reception ratio and its stability. The "Casino-lab" noise trace has been used in test case scenarios with a moderate to high degree of background noise, while the "Meyer-heavy" noise trace has been employed to produce scenarios with a very high degree of packet loss. Additionally, nodes can be uncontactable for short periods of time due to the stochastic behaviour in the packets reception ratio computed by the closest pattern matching (CPM) algorithm with the noise floor trace values as input.

At the Medium Access Control (MAC) layer, the contention-based Carrier Sense Multiple Access (CSMA) protocol with Collision Avoidance (CA) has been utilized. This is the default MAC protocol in TOSSIM which calculates backoff and preamble times according to the specifications of the CC2420 transceiver.

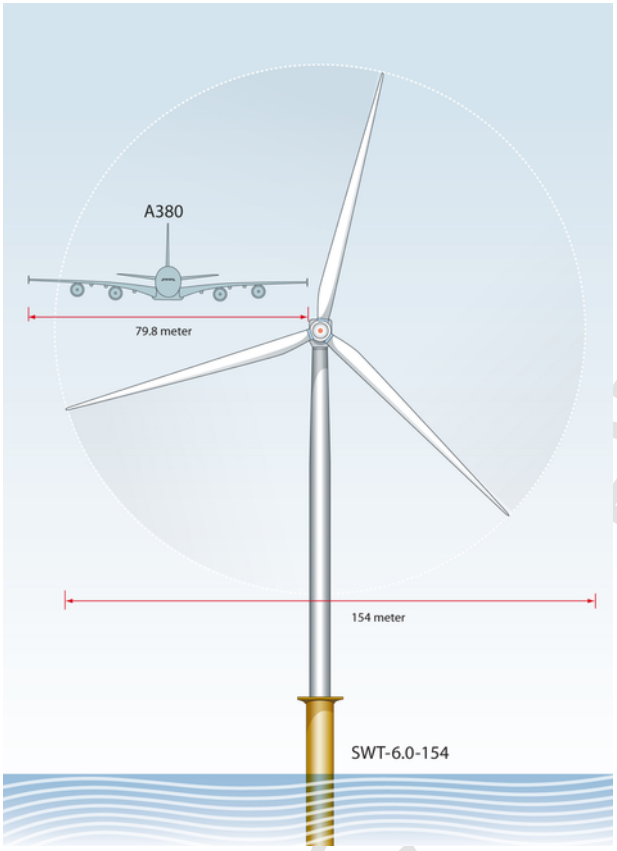

Fig. 10. Siemens D6 SWT-6.0–154 wind turbine generator (image courtesy of Siemens).

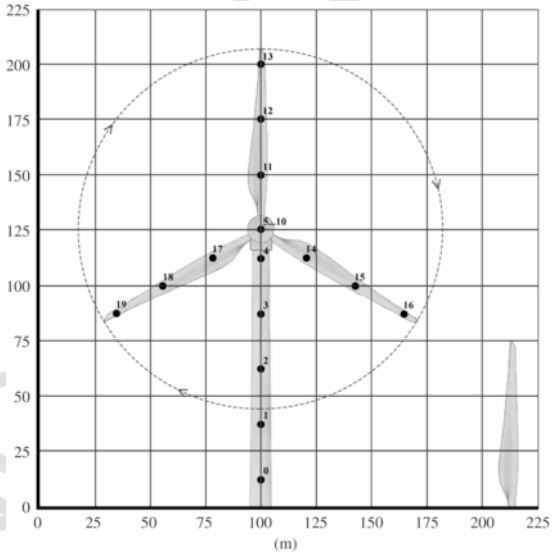

Fig. 11. 20-node wireless sensor network topology deployed on the siemens SWT-6.0–154 wind turbine.

4.2.2. Routing layer performance metrics

Next, the list of metrics used for the evaluation and comparison of the routing protocols are presented and defined.

1. Delivery Ratio (DR): The ratio of the number of unique application-layer packets which the source node is configured to send to the number of unique application-layer packets received at the destination node. The DR is aggregated, as a percentage, for all the participant source nodes in the network for each experiment.
2. Average End-to-End Delay (Latency): The average time packets take from the time they leave the application layer at the source node until they are received at the application layer at the destination node. The average is calculated from all packets sent at the application-layer in each experiment for a particular route (source to destination) in milliseconds.
3. Average Route Hops (Path Length): The average number of router nodes participating in the forwarding of the application-layer packet between the source and the destination node. The average is calculated using the number of hops stored in each packet for all packets received at the destination node from a particular source node.
4. Total Packets Sent (TPS): The number of packets sent by a node at the MAC layer.
5. Network Total Packets Sent (NTPS): The sum of the Total Packets Sent by all the nodes in the network.
6. Transceiver Energy Consumption: Energy consumed in mA by the transceiver of a node.
7. Transceiver Duty Cycle: The time the radio is active to the amount of time the radio is active and sleeping.

4.2.3. Experimental methodology

An application layer has been created for evaluation purposes which controls the sending and receiving of packets. In addition, an intermediate layer between the application layer and the routing protocols has been implemented. This layer acts as a bridge, adapting interfaces provided by the routing protocols to the communications functionality required at the application layer, and also serves to configure and initialize the routing protocols. For instance, CTP needs to be configured to appoint selected nodes to act as sinks such that their gradients are spread, typically at the beginning of the network operation. UMG can be configured for a node to spread its gradient at the beginning of the network operation or whenever the application requests it; a source node can also spread its gradient while requesting a destination node to spread its own gradient. The intermediate layer also implements an admission queue to store packets received from the application layer which are to be sent sequentially. The application layer controls the number of unique (by sequence) packets sent and received, as well as the latency and hop distance between two nodes.

The application layer payload in the packet contains the hop distance value which gets incremented at each intermediate node en route. The time when the packet is enqueued in the intermediate layer is also transported in the application payload. Latency is then calculated on reception with respect to the simulator clock. From all the unique packets received at the destination, an average of the latency time and the path length (hops) is computed. The metric Latency indicates the average time a packet takes to arrive at the destination node since it is enqueued in the intermediate layer at the source node. Thus, this metric indicates the capability of the routing protocol to deliver packets in each case scenario.

It has to be noted that UMG can operate with end-to-end acknowledgements enabled for every unique packet sent. On the other hand, CTP do not require an explicit acknowledgement to each packet, and thus do not need to wait for an end-to-end ACK packet from the destination. Instead, CTP sends packets when a route exists such that data packets are pushed towards the destination node in a stream fashion, employing per link acknowledgements and retransmissions. To facilitate fair comparison, UMG has been configured to operate with its end-to-end acknowledgement reliability mechanism disabled. Moreover, CTP employs a proactive regulated periodic beacon (the Trickle algorithm), to create and maintain quality routes towards a sink or a set of sink nodes. A packet sent from a source node does not have a destination address but rather is pushed towards the closest sink. On the contrary, UMG operates in a reactive manner and only creates and repairs routes when there is data to be transported between any two nodes in the network.

Two different types of traffic are generated: (1) Collection, and (2) Point-to-Point(P2P). In Collection, traffic flows from multiple points to one point. All 19 source nodes send a packet at each sending interval, in a quasi-simultaneous manner, towards the sink, i.e. Node 0. In Point-to-Point, communication occurs between any two nodes in the network. A number of routes between sensors placed at different areas are created to test the UMG protocol in scenarios where nodes communicate for cooperation purposes such as in-network processing tasks. The next 6 routes have been selected which create P2P traffic in between different components of the wind turbine (sender-destination): 19–16; 16–5; 13–3; 18–8; 15–9; 12–10. Note that CTP is not tested in scenarios of Point-to-Point traffic as it is a collection protocol configured to send data from sources to the nearest sink.

The sending interval impacts the end-to-end packet latency, the traffic network conditions and the path length. In terms of sending interval, three intervals have been tested: (1) 1000 ms interval (1 pck/s), (2) 2000 ms interval (1 packet every 2 s), (3) 500 ms interval (2 pck/s).

An evaluation of the protocols has also been carried out when nodes operate under Low-Power Listening (LPL) strategies where the radio duty cycles. The transceiver has been duty cycled for a period of 100 and 200 ms to analyse the effect on latency according to the following next strategies: (i) (20/100), 20 ms awake, next 80 ms in sleep mode, (ii) (30/100), 30 ms awake, next 70 ms in sleep mode, (iii) (40/200), 40 ms awake, next 160 ms in sleep mode, and (iv) (50/200), 50 ms awake, next 150 ms in sleep mode. The awake times for each cycle has been selected according to the conservative configuration of UMG and CTP, in

terms of the number of packet retrials and packet sending delays, such that high performance in latency and delivery ratio are achieved, while energy is saved as much as possible, with one configuration for all the various scenarios tested.

Table 2 shows the most representative experimental parameters varied in the simulations. Test scenarios have been created based on combinations of these parameters. A total number of 50 packets are sent from each source node during a simulation time of 300 s. The sending of packets starts randomly after a period of 10 s from the beginning of the simulation. Each experiment has been repeated 15 times. For the first set of experiments, where UMG is compared with CTP, the mean and 95% confidence intervals of the mean for each metric have been computed and are shown. For the remainder of the experiments, where the specific performance of UMG is evaluated, the median, maximum and minimum values are computed and plotted with error bars in the figures (to show the performance bounds of the protocol).

4.3. Experimental results and discussion

The UMG routing protocol performance has been compared with CTP in collection traffic scenarios. UMG has also been evaluated in collection and point-to-point traffic scenarios, including the role of the Inter-Cluster Gradient Formation (ICGF) algorithm in its performance. The performance of UMG operating with radio Low Power Listening (LPL) strategies is also discussed and its energy consumption is analysed.

4.3.1. Performance comparison of UMG and CTP

This section evaluates the performance of UMG in comparison with CTP in scenarios of collection traffic. In all the scenarios, the 19 nodes send packets towards the sink node, i.e. Node 0. A set of 12 scenarios have been created for comparison where the noise floor, rotor speed and sending interval have been varied. The first 6 scenarios have been configured with a moderate to high noise floor, i.e. Casino, while the last 6 scenarios employ a very high noise floor, i.e. Meyer. In each group of 6 different noise scenarios, the sending interval has been set to 2000, 1000 and 500 ms, and for each of these tests the rotor speed has been established at 10 and 6 rpm. Thus, the scenarios from left to right in the figures are organized from normal to very extreme conditions to facilitates the analysis of the protocols. The delivery ratio and network packets sent are plotted in Figs. 12 and 14 respectively. The route between node 19 and sink 0 has been selected for the study of latency and path length which are plotted in Figs. 13 and 15 respectively. This route is one of the longest in path length and high latency which communicates the tip of the blade with the lowest part of the tower

UMG clearly outperforms CTP in the majority of scenarios evaluated, and is superior to CTP (within the established margins of error) in the remaining scenarios. Fig. 12 shows that UMG maintains a high mean value of delivery ratio of 100% in all the scenarios with moderate to high noise, i.e. Casino, with 95% Confidence Intervals (CI) of less than 0.2% in all but one scenario, i.e. sending interval of 500 ms. CTP exhibits lower mean delivery ratios than UMG in these experimental scenarios with a

Table 2
Routing protocols simulation parameters.

Rotor speed	6 rpm; 10 rpm
Noise floor trace	Casino-lab; Meyer-heavy
Traffic type	Collection—Sink is node 0; Point-to-Point(P2P)
Sending interval	500 ms (2 pck/s); 1000 ms (1 pck/s); 2000 ms (1 pck/2 s)
Source packets sent	50
(Wake/sleep) LPL intervals	(20/100 ms); (30/100 ms); (40/200 ms); (50/200 ms)

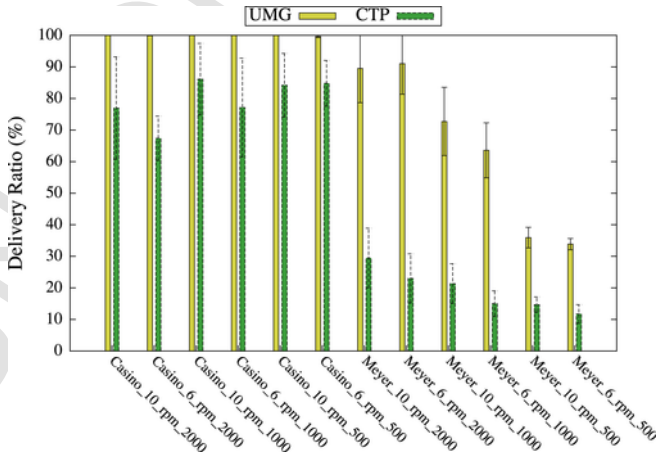

Fig. 12. UMG and CTP comparison on collection traffic (19 nodes sending to sink 0):—DELIVERY RATIO—[Mean, −95% CI, +95% CI].

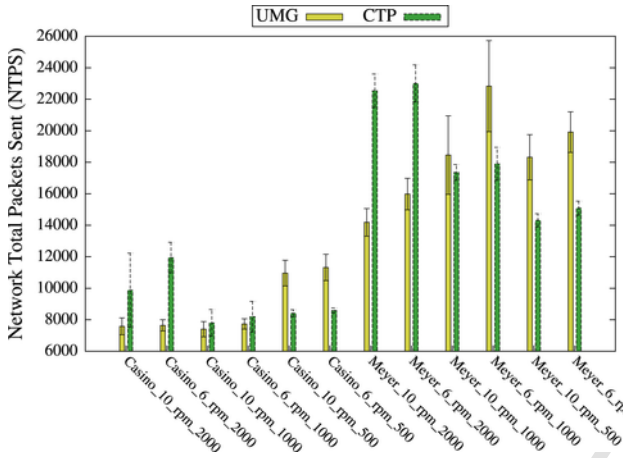

Fig. 14. UMG and CTP comparison on collection traffic (19 nodes sending to sink 0):—NETWORK TOTAL PACKETS SENT—[Mean, −95% CI, +95% CI].

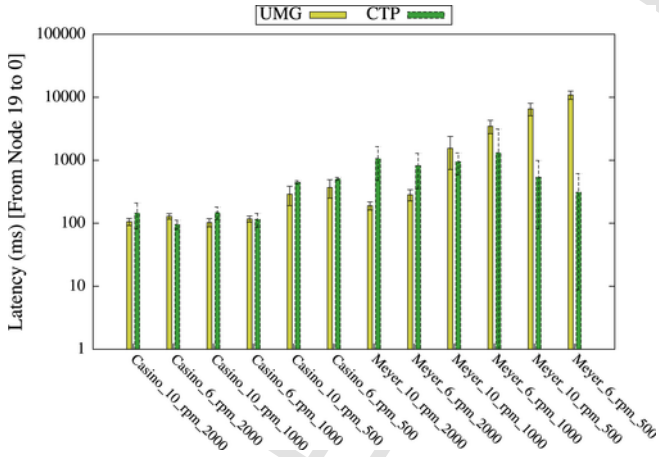

Fig. 13. UMG and CTP comparison on collection traffic (node 19 sending to sink 0):—LATENCY—[Mean, −95% CI, +95% CI].

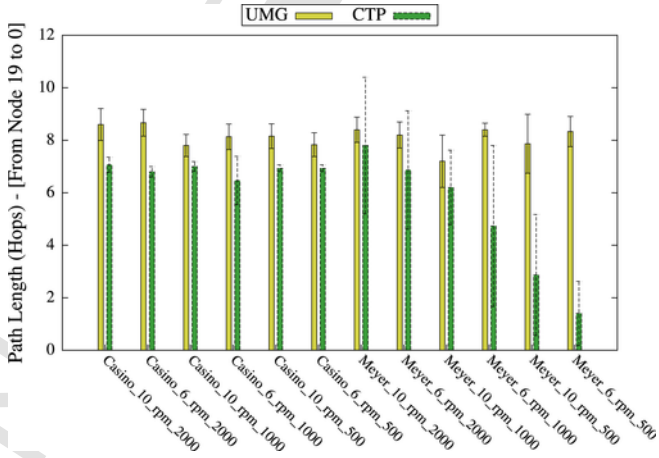

Fig. 15. UMG and CTP comparison on collection traffic (node 19 sending to sink 0):—PATH LENGTH—[Mean, −95% CI, +95% CI].

markedly broader 95% CI indicating greater instability in the delivery of packets. This scenario arises due to the mobility of the nodes placed on the blades and the lack of multiple alternate routing nodes through which to forward data. In scenarios of very high noise, i.e. Meyer, the mean value of the delivery ratio for UMG reaches at least 90% in scenarios with moderate sending in-

tervals, i.e. every 2000 ms, with 95% CI of around 10% achieving throughput of 100%. However the mean value drops markedly when the sending interval is lower and the packet injection rate is higher. The lower mean values exhibited by UMG with Meyer noise indicates that, as expected, the high noise in combination with the low number of neighbours per node diminishes the performance of the protocol. In these scenarios, CTP's delivery ratio also decreases significantly to mean values of 30%–12%, again indicating that the adverse networking conditions created with the Meyer noise profile serve to stress the performance of the protocols. Overall, the results show that there is not a significant difference in UMGs delivery ratio (for the same scenarios) with rotational speeds of 6 or 10 rpm.

The total number of packets sent in the network for each scenario (see Fig. 14) shows the impact that the Meyer noise floor has in the network as compared to the Casino background noise. While CTP sends less network total packets than UMG in some scenarios, the delivery ratio of CTP is always lower in these cases. Inspection of the figures shows that UMG employs a lower number of packets sent than CTP.

In terms of the average latency of the packets transmitted in the route from node 19 to sink 0, plotted in Fig. 13, both protocols deliver the packet in around 100 ms in scenarios of Casino noise floor and sending intervals of 1000 and 2000 ms, whereas a lower sending interval of 500 ms increases the delivery latency (values remain lower than 1 s). In scenarios of very high noise, Meyer, lowering the sending interval has a negative impact on the protocols, where in the worst cases UMG reaches upper mean latencies of 10 s. CTP latency values can be lower in these types of high noise scenarios, although we caution that the delivery ratios are very low and the network congestion varies providing opportunities for lower latencies. Overall, these scenarios confirm the challenging network conditions created and the robust capabilities of UMG to route packets. No significant difference in latency is noted when the speed of the rotor changes, despite some variation in scenarios of Meyer noise and variation (1000 and 2000 ms) of the sending intervals. Note that UMG has been configured with random delays in the forwarding of packets to improve the performance when using Low Power Listening strategies and thus the latency values could be further optimized. In addition, latency values need to be studied considering path length in Fig. 15. UMG employs 1–3 hops more than CTP in the delivery of packets due to the searching for potential new routes at the nacelle when one-hop communication fails. Longer path lengths are also attributable to the Inter-Cluster Gradient Formation mechanism (see Section 3.5.1) which deals with mobility by creating routes between (intra) and within (inter) stable areas. While some resultant routes can be longer, this does not have a significant impact on the latency values and has a positive effect in the delivery ratio. Note that CTP produces long path lengths of more than 10 hops in some experimental scenarios due to its loop prevention mechanism which enable a packet to travel the loop until the route is repaired.

4.3.2. Performance analysis of UMG inter-cluster gradient formation algorithm

The Inter-Cluster Gradient Formation (ICGF) algorithm presented in Section 3.5.1 is now evaluated to demonstrate its impact on the performance of UMG as a mechanism for creating areas and selecting stable routes, while avoiding transient/mobile nodes acting as routers. The same configuration and set of collection traffic scenarios employed in the previous section has been used to study and compare the performance of UMG when the ICGF mechanism is enabled (ICGF on) and disabled (ICGF off). The median, minimum and maximum values of the evaluation metrics are calculated.

In terms of delivery ratio, Fig. 16 shows that for scenarios of moderate to high noise, UMG operating with the ICGF algorithm enabled is more reliable. producing values for median, maximum and minimum close to 100%. In these same scenarios, with the ICGF algorithm disabled, the results show higher disparities between max and min values, particularly in scenarios with lower sending interval e.g. 500 ms. For high noise experiments, Meyer, the median value of the delivery ratio is usually higher when ICGF is enabled. However the variability between min and max values is also higher. This indicates that, in very high

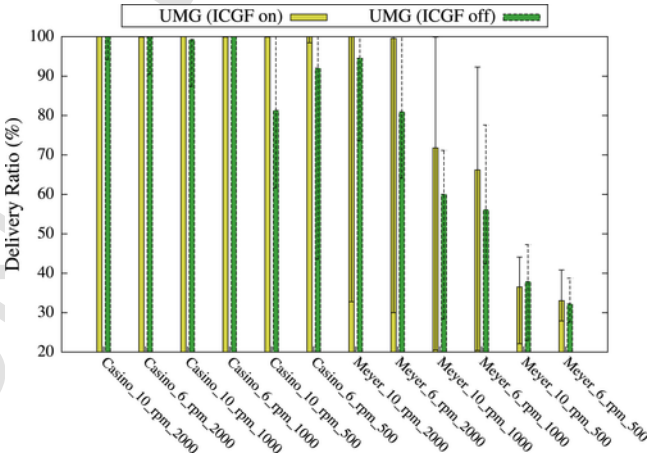

Fig. 16. UMG performance with ICGF enabled/disabled on collection traffic (19 nodes sending to sink 0):—DELIVERY RATIO—[Median, Min, Max].

noise scenarios, an opportunistic approach which selects neighbour nodes (even if they are transient) as routers can be beneficial. Note that these scenarios are designed for stress testing of the protocols and the delivery ratios in these scenarios are consequentially lower than in realistic operational scenarios.

When studying the behaviour of UMG operating with and without ICGF with regards to latency. Fig. 17 indicates that, in general, by enabling the ICGF algorithm similar or lower median latencies are achieved. On the other hand, when the ICGF is disabled the disparity between min and max values of latency increases. Note that, in two scenarios of high noise the minimum value of latency is zero; this indicates that for a particular test the packet did not leave the origin node due to the high noise and low density of neighbours capable of acting as routers. The mean number of network total packets sent, plotted in Fig. 18, is generally lower when the ICGF mechanism is enabled, whilst requiring high number of packets when achieving maximums for the majority of scenarios. This indicates that the protocol employs more network packets when routes are unstable, as might be expected.

Finally, the impact of the ICGF algorithm in the selection of stable routes and the avoidance of transient nodes is demonstrated through the path length in Fig. 19. For most cases, the mean values of path length is 1–2 hops higher when ICGF is active exhibiting a higher disparity between max and min values. This results indicate that the ICGF mechanism may select longer routes at the benefit of improving reliability, latency and packet overhead.

4.3.3. Performance analysis of UMG in collection traffic scenarios

In addition, the UMG routing protocol has been evaluated when operating with Low-Power Listening (LPL) strategies. UMG has outperformed CTP in all LPL scenarios and across different wind turbine rotor speed. A selection of the available experimental data is presented. Specifically 6 scenarios with a speed of 10 rpm discussed above have been selected to analyse the per-

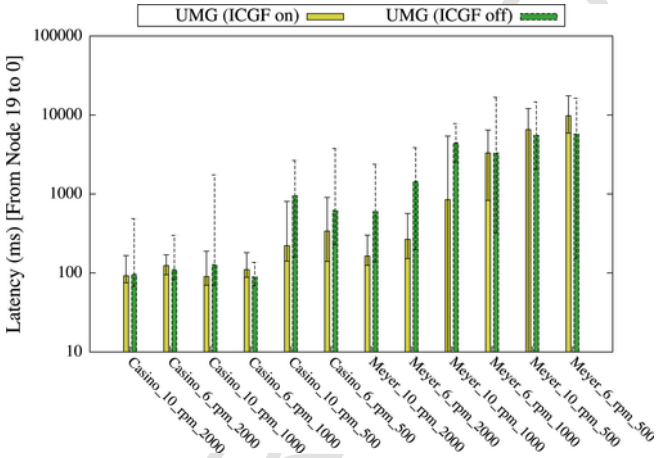

Fig. 17. UMG performance with ICGF enabled/disabled on collection traffic (19 nodes sending to sink 0):—LATENCY—[Median, Min, Max].

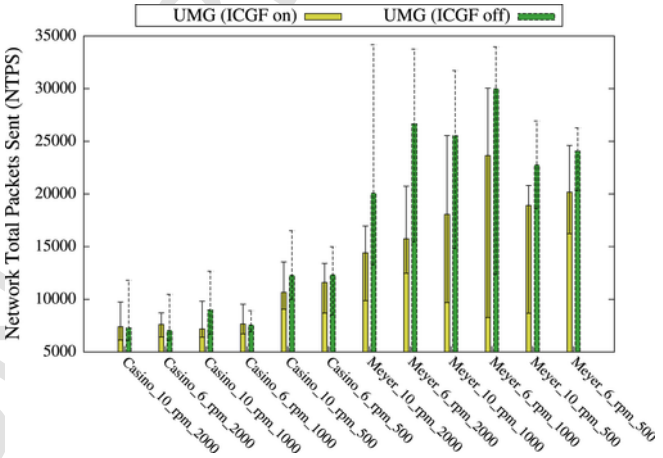

Fig. 18. UMG performance with ICGF enabled/disabled on collection traffic (19 nodes sending to sink 0):—NETWORK TOTAL PACKETS SENT—[Median, Min, Max].

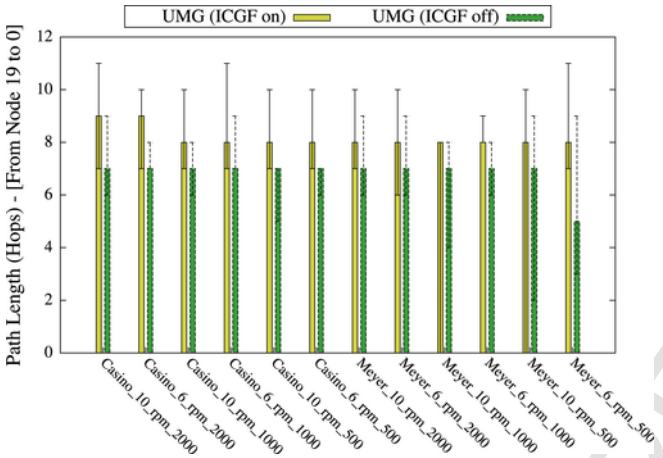

Fig. 19. UMG performance with ICGF enabled/disabled on collection traffic (19 nodes sending to sink 0):—PATH LENGTH—[Median, Min, Max].

formance of UMG when the transceiver has a duty cycle of: Radio On, 30/100, 20/100, 50/200 and 40/200 (plotted in this order). The low-power layer employs a dynamic sleeping mechanism explained in Section 3.6. The delivery ratio and network total packets sent are plotted in Figs. 20 and 22 respectively, while the latency and path length for the route from node 19 to sink 0 are illustrated in Figs. 21 and 23 respectively.

Fig. 20 shows that most of the duty cycle strategies produce a high median delivery ratio value of more than 95% in scenarios of moderate to high noise, i.e. Casino, with a moderate sending interval of 2000 ms. When the sending interval is decremented to 1000 ms, only the LPL strategy with 30/100 duty cycle remains at the same high median delivery ratio value of 100%, with the strategy 50/200 dropping down to 95% and the other two strategies 20/100 and 40/200 decreasing to 89% and 80% respectively. In addition, their maximum values peak at 98%–100%. However it should be noted that the minimum values can drop to less than 60% even with the most reliable LPL strategy, i.e. the 30/100 duty cycle, for a minority of experiments. In this strategy the transceiver wakes up for a sufficient time to send and receive packets, and to maintain the network awake if activity occurs, while at the same time it does not sleep long so packets are not missed and packets do not overflow the queues. Nevertheless, the traffic in the network highly increases for a test where 19 nodes start sending at closer times in each sending interval, despite some limited randomization, which gets exacerbated when the sending interval is lowered. The rationale of having all the nodes send packets at closer times in each sending interval seeks to test the performance of the protocol when the network is stressed, however it can also generate scenarios where a good proportion of the nodes inject packets at the same time thus increasing the traffic load. Moreover, if some packets cannot progress to the next node, e.g. if this node is sleeping or the noise floor is interfering in the communication, then packets can overflow the queue and be discarded after some trials and local repair processes. In particular for these scenarios, where all nodes have the same destination, sink 0, and the number of forwarders for each node is sometimes limited to 1 neighbour, the traffic can be subject to bottlenecks. This effect can also be seen in the median values in the delivery ratio of the scenario of Casino noise and a sending interval of 500 ms, as well as in all the scenarios

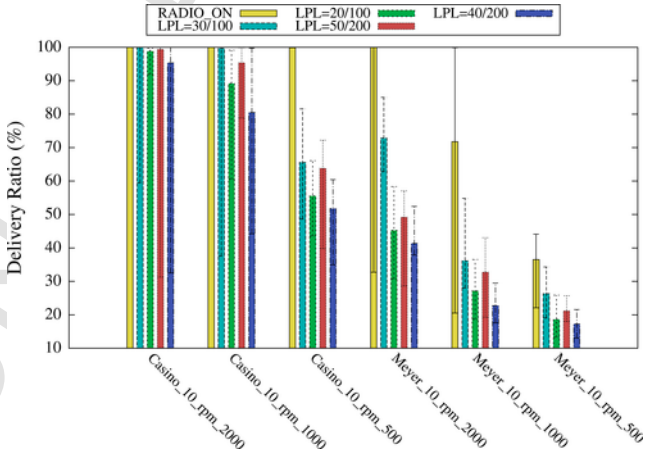

Fig. 20. UMG performance with LPL strategies on collection traffic (19 nodes sending to sink 0):—DELIVERY RATIO—[Median, Min, Max].

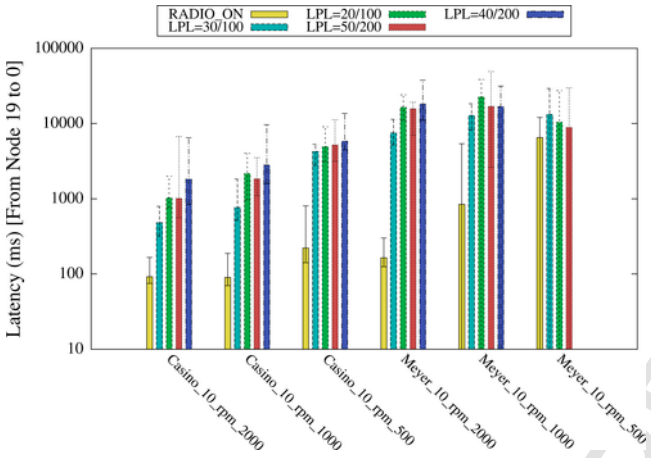

Fig. 21. UMG performance with LPL strategies on collection traffic (node 19 sending to sink 0):—LATENCY—[Median, Min, Max].

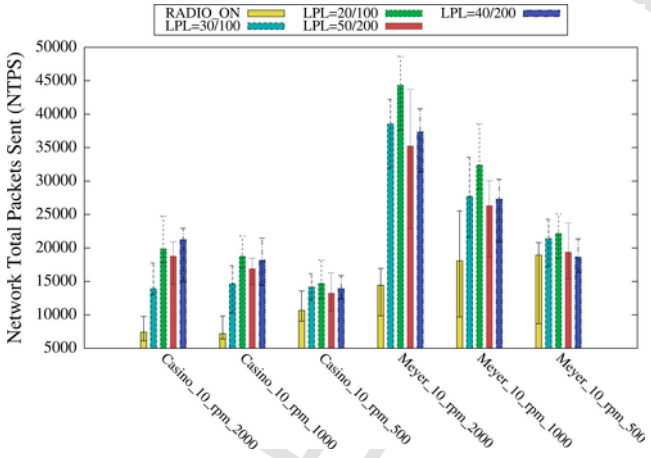

Fig. 22. UMG performance with LPL strategies on collection traffic (19 nodes sending to sink 0):—NETWORK TOTAL PACKETS SENT—[Median, Min, Max].

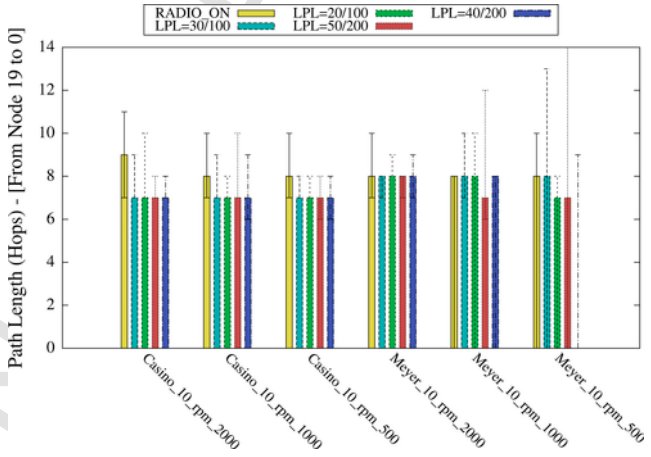

Fig. 23. UMG performance with LPL strategies on collection traffic (node 19 sending to sink 0):—PATH LENGTH—[Median, Min, Max].

with the very high Meyer noise floor. In addition, when looking at the packets sent in Fig. 22, it can be observed that LPL strategies require a higher number of packet to be sent in the network which is also aligned with the delivery ratio, i.e. the higher the delivery ratio the higher the number of network total packets sent. This is due to the dynamic sleeping mechanism which wakes up to send a packet and keeps on trying, with randomized delays in between transmissions, until the selected neighbour wakes up and acknowledge its reception, or until a local repair is successful. This greatly increases the overhead in sending packets at the benefit of saving energy. In terms of delivery ratio and network total packets sent, the most suitable strategy is 30/100, followed by 50/200. This is explained by the fact that these two strategies spend slightly higher time in the active state than the other two in each corresponding cycle. However, and as expected, when we inspect the average latency for the route 19 to 0 in Fig. 21, the lowest latencies correspond to the duty cycle strategy 30/100, which for the first two experiments stay below the 1 s limit thus indicating that the network does not suffer from excessive traffic load. This result also confirms the effect of the sending interval and the high noise floor in increasing the traffic load and network congestion which reflects on latencies higher than 5 s. Finally, the path length in Fig. 23 indicates that in scenarios of moderate to high noise and sending interval, the delay introduced with the LPL strategies is beneficial when descending the gradient towards the sink in such a way as to minimize the number of local repairs, which also contributes to decrease the number of hops. This suggests that the gradients are formed properly and that the duty cycling of nodes in conjunction with the delay in sending packets reduces the wireless medium contention at the cost of higher latencies and lower delivery ratios.

4.3.4. Performance analysis of UMG in point-to-point traffic scenarios

This section evaluates the performance of UMG in scenarios of point-to-point traffic where 6 routes have been selected. Each of the selected sender and destination nodes enable testing of communication between different areas of the wind turbine structure (sender-destination): (19–16; 16–5; 13–3; 18–8; 15–9; 12–10). The homologous 6 scenarios employed in evaluating LPL strategies for collection traffic in the previous section, where the rotor speed is 10 rpm, have been employed as the results were similar to those where the rotor moves at 6 rpm. Similarly, the same LPL duty cycle strategies have been evaluated. Again, the scenarios from left to right in the figures are organized from moderate to very extreme conditions to facilitate the analysis of the protocol performance.

Fig. 24 shows higher delivery ratios than the homologous results for collection traffic in Fig. 20. This is expected due to the higher degree of network traffic distribution in point-to-point traffic scenarios, which is a consequence of having multiple destinations and a lower number of sources. UMG achieves 100% median delivery ratios in all scenarios when the radio is active all the time, as well as minimum delivery ratios higher than 98%. The 30/100 LPL strategy produces high delivery ratios close to those achieved when the radio is always on, specifically in scenarios of moderate noise, Casino, and the scenario of very high noise, Meyer, with a moderate sending interval of 2000 ms. Additionally, it needs to be noted that in scenarios of Casino noise floor all the LPL strategies achieve at least 97% delivery ratio, although the 30/100 and 20/100 LPL strategies achieve the best minimum delivery ratios thus suggesting the adoption of duty cycles with low sleeping times. Decreasing the sending interval has also a negative impact on LPL strategies with higher sleeping times in moderate to high noise scenarios. However in very high noise scenarios, where lowering the sending interval contributes to increase the peak traffic, the median and minimum delivery ratios for the LPL strategies decay as the sending interval lowers and the 30/100 LPL strategy is postulated as the best one with the 20/100 LPL strategy performing worse than the 50/200 one. This indicates that the protocol is sensitive to the traffic in the network, as it was seen in extreme scenarios of collection traffic, and that lower sleeping times need to be selected for LPL strategies, while at the same time adapting the awake time to a value in line with the packet retransmission delay and the maximum retransmissions of the protocol.

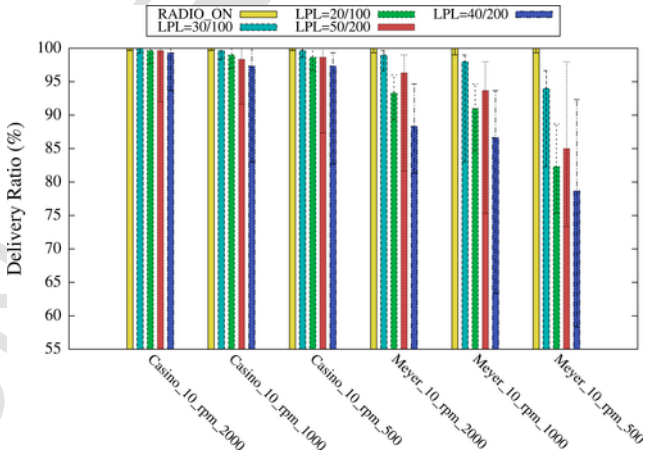

Fig. 24. UMG performance with LPL strategies on P2P traffic (6 senders to 6 destinations):—DELIVERY RATIO—[Median, Min, Max].

The cost in packets sent of using the proposed LPL strategies over having the radio always active can be seen in Fig. 26 where again the 30/100 LPL strategy produces the best results when duty cycling, doubling the amount of network total packets sent when compared to results when the radio is on in scenarios of Casino noise.

UMG's point-to-point performance in terms of latency is plotted in Fig. 25 for the route between node 19, placed at the tip of one of the blades, and node 16, placed at the tip of another blade. The path lengths achieved for this route are displayed in Fig. 27 showing median and minimum lengths of 5–6 hops with maximums of 7 and 8 hops due to local repair activities. When the radio is on, the latency stays under 100 ms even in scenarios of very high noise floor, Meyer. When employing LPL strategies, the latencies increase as expected, where the 30/100 LPL strategy produces the best values with median latencies below 1 s for most of the scenarios. This proves the efficacy of UMG in delivering packets under extreme conditions. Nevertheless, very high noise can have a negative impact on a few experiments, indicated by high maximum values of latency, specially in strategies with the longest sleeping times which produce latencies of more than 1 s. Overall, the use of UMG with the 30/100 LPL strategies in moderate to high noise scenarios produces results in line with those when the radio is always on, and its performance in very high noise scenarios decays slightly when the sending interval lowers.

4.3.5. Analysis of energy consumption and duty cycle

One of the realistic scenarios has been selected to analyse the energy consumed by each node in the network as well as the dynamically changing duty cycle for three selected nodes. The scenario chosen is the moderate to high noise floor, Casino, with a rotor speed of 10 rpm and a sending interval of 2000 ms. Both for collection and point-to-point traffic, one of the tests for the scenario has been analysed when the transceiver is always on and when operating using the 20/100 LPL strategy. In this scenario, data traffic occurs in the network during the first 110 s, as 50 packets are sent every 2000 ms, and the nodes start sending after an initial delay of 10 s. The experiment has a duration of 300 s. With the 20/100 duty cycle strategy, the minimum energy

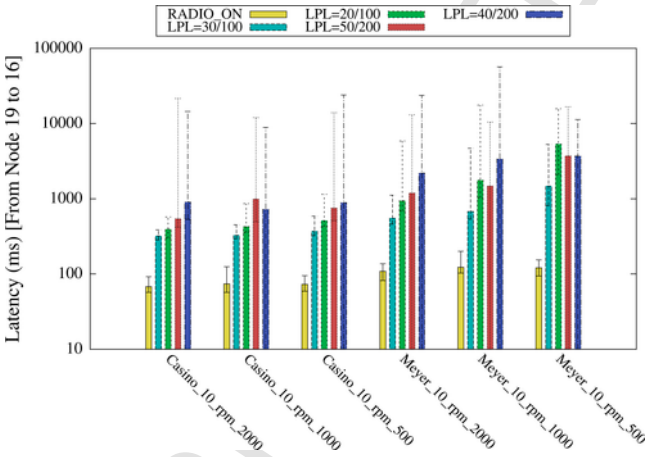

Fig. 25. UMG performance with LPL strategies on P2P traffic (node 19 sending to node 16):—LATENCY—[Median, Min, Max].

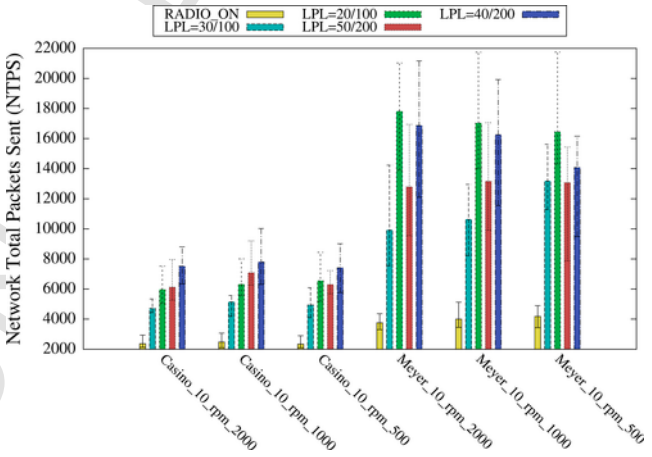

Fig. 26. UMG performance with LPL strategies on P2P traffic (6 senders to 6 destinations):—NETWORK TOTAL PACKETS SENT—[Median, Min, Max].

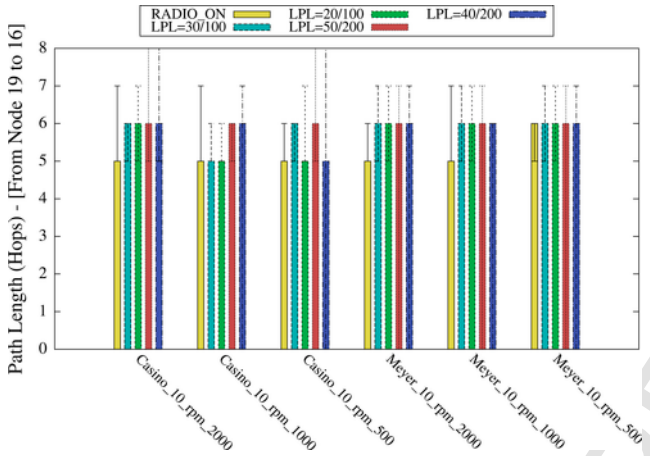

Fig. 27. UMG performance with LPL strategies on P2P traffic (node 19 sending to node 16):—PATH LENGTH—[Median, Min, Max].

consumption that can be achieved is 20% when the radio is awake in such proportion for the whole experiment. This is predominantly achieved when there is not much traffic in the network and it is set as the lower limit to which the nodes duty cycle tends to adapt when there is no traffic.

For the collection traffic scenario, the energy consumed by each node in the network is displayed in Fig. 28 while the duty cycle for nodes 1, 5 and 19 is plotted in Fig. 30. When the radio is always awake, the consumption of all the nodes is the same, close to 6000 mA for the 300 s period. Employing the 20/100 LPL strategy, the energy saving can fluctuate between 78% and 70% depending on the node. Nodes 0–4 show a higher energy consumption than the rest of the nodes as all packets need to be forwarded through them to reach the sink node 0. However, those nodes placed at the nacelle, nodes 5–10 have different energy consumption depending on their participation as routers. For the rest of the nodes placed at the blades the energy consumption is low, where the lowest values are given by those sensors at the tip of the blades as their participation in routing activities is limited. In terms of the duty cycle, node 1 produces high percentages, close to 50%, when the radio is regularly awake for routing purposes during the first 110 s which starts to tend towards the 20% value when the traffic lowers in the network. This is expected as node 1 is a busy node acting as a main router for the sink node. This can be compared with node 5 which does not get elected as a router until second 60, as it is placed at the nacelle where more nodes are available. In the same vein, node 19 placed at the tip of the blade does not participate in the routing activities and consequently its radio duty cycle exhibits values close to 20% for the majority of the experiment duration.

For the point-to-point (P2P) traffic scenario, the energy consumed by each node in the network is displayed in Fig. 29 while the duty cycle for nodes 1, 5 and 19 is plotted in Fig. 31. Contrary to the collection traffic experiment, the traffic is balanced among routes between source and destination nodes which connect: (i) the same blade, (ii) two blades, (iii) blades with nacelle and (iv) blade with the top of the tower. Consequently, when using the 20/100 LPL strategy, nodes at the tower exhibit lower consumption as they do not participate much in the routing process. Nodes at the blades, and some nodes at the nacelle, consume

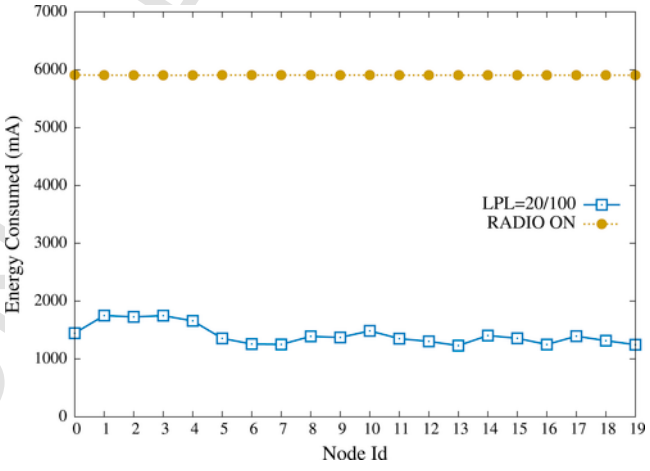

Fig. 28. UMG's transceiver energy consumption—collection traffic scenario (10 rpm, Casino, 2000 ms).

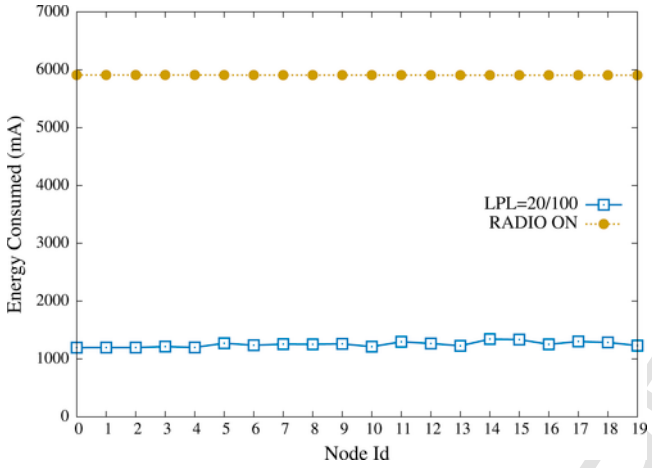

Fig. 29. UMG's transceiver energy consumption—P2P traffic scenario (10 rpm, Casino, 2000 ms).

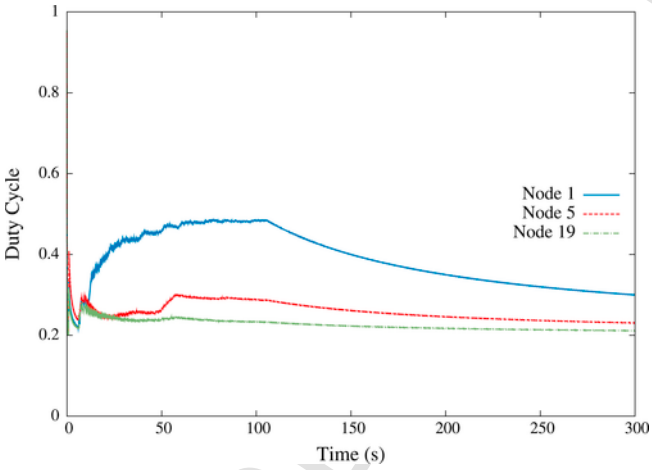

Fig. 30. UMG's transceiver duty cycle—collection traffic scenario (10 rpm, Casino, 2000 ms).

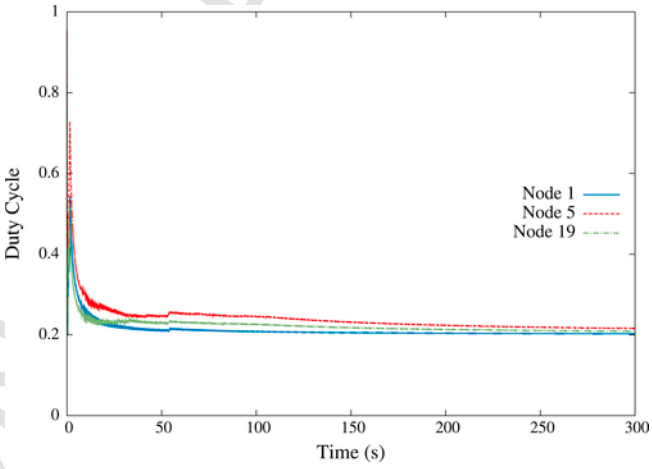

Fig. 31. UMG's transceiver duty cycle—P2P traffic scenario (10 rpm, Casino, 2000 ms).

slightly more energy. Most of the nodes save a similar amount of energy which varies between 79% and 77% with respect to the scenario when the radio is always on. When looking at the duty cycle of the selected nodes, it is clear to see that these nodes do not participate much in the routing process and its duty cycle quickly adapts and stay at values close to 20%. For this scenario, node 1 quickly converges towards the 20% duty cycle (see Fig. 31) as it does not route packets.

4.3.6. Overall evaluation and operating guidelines of UMG

The UMG routing protocol has been tested under a variety of scenarios in a mobile network topology for wind turbine monitoring. Moderate to very high interference and traffic conditions were simulated to study the limitations in the performance of the protocol in a network characterized by having a low degree of neighbours for routing activities. Under these conditions, UMG has outperformed the Collection Tree Protocol (CTP) in all collection traffic scenarios. This fact indicates the routing effectiveness and reliability of UMG, specially when taking into account that CTP is the de-facto collection protocol in WSNs and that it has been proven to be highly reliable. This also indicates the extreme conditions of the scenarios created. When designing wind turbine monitoring applications and deploying wireless sensor networks, some of the scenarios tested should be considered as of unreliable and unfeasible network condition.

Consequently, some guidelines for the best operating conditions of UMG can be deduced from the evaluation. First, UMG operating with the adaptive Low Power Listening layer has produced good performance in realistic scenarios in terms of reliability and latency with energy savings of nearly 80%. LPL strategies with low sleeping times are preferred as the protocol becomes more reliable and reacts quicker to changes. Second, applications should be configured to process data and send digested information while avoiding streaming traffic. The number of packets injected in the network should be controlled as to avoid wireless medium congestion and queue overflow. An interval of 2000 ms, i.e. one packet every two seconds, has produced good results in most of the scenarios. It needs to be considered that in collection scenarios all 19 source nodes started injecting packets quasi-simultaneously which should seldom occur in a real deployment. Leveraging UMG's versatile communication system, the sending interval can be regulated according to traffic in different areas. Third, UMG can improve its performance when some nodes are mobile by defining areas in supervised deployments, such as in wind turbine structures where its performance is not impacted by the speed and direction of the moving nodes. Finally, UMG can perform collection, point-to-point and point-to-multipoint routing activities simultaneously, enabling data collection, network reconfiguration and in-network processing from any point in the network.

5. Conclusions

The Ubiquitous Mobile Gradient (UMG) routing protocol presented has been designed to provide versatile communication support to higher layers in the domain of wireless sensor networks for wind turbine monitoring. However the protocol can be applied to a variety of applications in the area of WSNs. UMG is a gradient-based reactive routing protocol integrating a collection of mechanisms to enhance the efficiency and reliability of the routing process which (i) provides reliable point-to-multipoint, multipoint-to-point and point-to-point communications, (ii) combines address-centric and data-centric routing concepts to provide service advertisement and discovery to higher layers, (iii) employs a backoff-based reliable controlled flooding mechanism for the progressive formation of gradients according to the hop distance, (iv) allows for the coexistence of multiple gradient updates from the same sink node, thus enabling nodes which could not get updated to participate in the forwarding process without incurring in loop formation, (v) provides mechanisms for navigating the gradient in both directions, and (vi) integrates mechanisms to tolerate mobility by enabling the creation of virtual clusters/areas for routing control purposes.

UMG has been successfully tested in the simulator monitoring one of the largest offshore wind turbine structures using a wireless sensor network composed of 20 nodes under different conditions of noise, traffic and rotor speed. Testing has validated the performance of the protocol in scenarios of data collection and point-to-point traffic with different traffic loads. UMG has been compared with CTP, the de-facto collection routing protocol in WSNs, and has proven to outperform CTP in all the proposed scenarios. In addition, Low Power Listening strategies have been tested with UMG achieving high delivery ratios and low latencies in scenarios with moderate noise, extending the battery life of nodes by at least a factor of 5, depending on the traffic generated by the application. Finally, UMG has been successfully validated as a suitable routing solution in WSNs for wind turbine structure monitoring where data collection and in-network processing activities are required. Future works will seek to extend the battery life of the sensor network with protocol-adaptive low power listening strategies and the UMG routing protocol will be tested in the simulator using different mobility patterns and on a real wind turbine structure.

Acknowledgements

This work has been partially supported by the European Commission through the FP7 ITN SYSWIND project (Grant No: 238325) and the H2020 EID ICONN project (Grant No: 675659).

References

[1] E.S. Carbajo, R.S. Carbajo, C. Mc Goldrick, B. Basu, Asdah: An automated structural change detection algorithm based on the Hilbert–Huang transform, Mech. Syst. Signal Process. 47 (1) (2014) 78–93.
[2] P. Tchakoua, R. Wamkeue, M. Ouhrouche, F. Slaoui-Hasnaoui, T.A. Tameghe, G. Ekemb, Wind turbine condition monitoring: State-of-the-art review, new trends, and future challenges, Energies 7 (4) (2014) 2595–2630.
[3] Z. Daneshi-Far, G. Capolino, H. Henao, Review of failures and condition monitoring in wind turbine generators, In: XIX International Conference on Electrical Machines, (ICEM), IEEE, 2010, pp. 1–6.
[4] R.S. Carbajo, A. Staino, K.P. Ryan, B. Basu, C. Mc Goldrick, Characterisation of wireless sensor platforms for vibration monitoring of wind turbine blades, in: 22nd Irish Signals and Systems Conference, ISSC, 2011, pp. 171–176.
[5] R.A. Swartz, J.P. Lynch, S. Zerbst, B. Sweetman, R. Rolfes, Structural monitoring of wind turbines using wireless sensor networks, Smart Struct. Syst. 6 (3) (2010) 183–196.
[6] G. Kilic, M.S. Unluturk, Testing of wind turbine towers using wireless sensor network and accelerometer, Renew. Energy 75 (2015) 318–325.
[7] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Comput. Netw. 38 (4) (2002) 393–422.
[8] R.S. Carbajo, E.S. Carbajo, B. Basu, C. McGoldrick, Real-time autonomous structural change detection onboard wireless sensor platforms, In: Key Engineering Materials, Vol. 569, Trans. Tech. Publ., 2013, pp. 970–977.
[9] S. Hedetniemi, S. Hedetniemi, A. Liestman, A survey of gossiping and broadcasting in communication networks, Networks 18 (4) (1988) 319–349.
[10] J. Al-Karaki, A. Kamal, Routing techniques in wireless sensor networks: a survey, IEEE Wirel. Commun. 11 (6) (2004) 6–28.
[11] C. Perkins, E. Royer, Ad-hoc on-demand distance vector routing, in: Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, Vol. 2, 1999, pp. 90–100.
[12] D. Johnson, D. Maltz, In: Dynamic Source Routing in Ad Hoc Wireless Networks, Kluwer International Series in Engineering and Computer Science, 1996, pp. 153–179.
[13] C. Perkins, P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, ACM SIGCOMM Comput. Commun. Rev. 24 (4) (1994) 234–244.
[14] C. Intanagonwiwat, R. Govindan, D. Estrin, Directed diffusion: a scalable and robust communication paradigm for sensor networks, in: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, 2000, pp. 56–67.
[15] W. Heinzelman, J. Kulik, H. Balakrishnan, Adaptive protocols for information dissemination in wireless sensor networks, in: Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, 1999, pp. 174–185.
[16] D. Braginsky, D. Estrin, Rumor routing algorithm for sensor networks, in: Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, 2002, pp. 22–31.
[17] C. Schurgers, M. Srivastava, Energy efficient routing in wireless sensor networks, in: IEEE Military Communications Conference, MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force, Vol. 1, 2001, pp. 357–361.
[18] R. Poor, Gradient routing in ad hoc networks, Tech. Rep., Media Laboratory, Massachusetts Institute of Technology Cambridge, MA, 2000.
[19] D. Estrin, R. Govindan, J. Heidemann, S. Kumar, Next century challenges: Scalable coordination in sensor networks, in: Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, 1999, pp. 263–270.
[20] F. Ye, A. Chen, S. Lu, L. Zhang, A scalable solution to minimum cost forwarding in large sensor networks, in: Tenth International Conference on Computer Communications and Networks, 2001, pp. 304–309.
[21] F. Ye, S. Lu, L. Zhang, Gradient broadcast: A robust, long-lived large sensor network, Tech. Rep., UCLA Internet Research Lab Papers, 2001.
[22] F. Ye, G. Zhong, S. Lu, L. Zhang, Gradient broadcast: A robust data delivery protocol for large scale sensor networks, Wirel. Netw. 11 (3) (2005) 285–298.
[23] K. Jaffres-Runser, C. Comaniciu, A probabilistic interference and energy aware gradient broadcasting algorithm for wireless sensor networks, in: 3rd International Symposium on Wireless Pervasive Computing, ISWPC 2008, 2008, pp. 1–5.
[24] K. Jaffres-Runser, C. Comaniciu, J. Gorce, R. Zhang, U-GRAB: a utility-based gradient broadcasting algorithm for wireless sensor networks, in: IEEE Military Communications Conference, MILCOM 2009, 2009, pp. 1–7.
[25] K. Jaffrès-Runser, C. Comaniciu, J. Gorce, Interference and Congestion Aware Gradient Broadcasting Routing for Wireless Sensor Networks, Tech. Rep., 2010. Arxiv Preprint arXiv:0902.0746.
[26] L. Xia, X. Chen, X. Guan, A new gradient-based routing protocol in wireless sensor networks, Embedded Softw. Syst. (2005) 318–325.
[27] Q. Fang, J. Gao, L. Guibas, V. de Silva, L. Zhang, GLIDER: Gradient landmark-based distributed routing for sensor networks, in: INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Vol. 1, 2005, pp. 339–350.
[28] J. Lim, K. Shin, Gradient-ascending routing via footprints in wireless sensor networks, in: Proceedings IEEE Real-Time Systems Symposium, RTSS'05, 2005, pp. 298–307.
[29] B. Bloom, Space/time trade-offs in hash coding with allowable errors, Commun. ACM 13 (7) (1970) 422–426.
[30] L. Fan, P. Cao, J. Almeida, A. Broder, Summary cache: a scalable wide-area web cache sharing protocol, IEEE/ACM Trans. Netw. 8 (3) (2000) 281–293.
[31] T. Schoellhammer, B. Greenstein, D. Estrin, Hyper: A routing protocol to support mobile users of sensor networks, Tech. Rep., Center for Embedded Network Sensing (CENS), Univ. of California, 2006.
[32] J. Wu, P. Havinga, Reliable cost-based data-centric routing protocol for wireless sensor networks, in: Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, SNPD 2006. 2006, pp. 267–272.
[33] P. Huang, H. Chen, G. Xing, Y. Tan, SGF: a state-free gradient-based forwarding protocol for wireless sensor networks, ACM Trans. Sens. Netw. (TOSN) 5 (2) (2009) 1–25.
[34] N. Khan, Z. Khalid, G. Ahmed, GRAdient cost establishment (GRACE) for an energy-aware routing in wireless sensor networks, EURASIP J. Wirel. Comm. Netw. 2009 (2009) 1–15.
[35] Y. Guo, Z. Xu, C. Chen, X. Guan, DGR: dynamic gradient-based routing protocol for unbalanced and persistent data transmission in wireless sensor and actor networks, J. Zhejiang Univ.-Sci. C 12 (4) (2011) 273–279.
[36] J. Faruque, A. Helmy, TABS: Link loss tolerant data routing protocol for multi-hop wireless sensor networks, in: 2010 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, 2010, pp. 11–18.
[37] R. Fonseca, O. Gnawali, K. Jamieson, S. Kim, P. Levis, A. Woo, TEP 123: The Collection Tree Protocol, 2006, http://www.tinyos.net/tinyos-2.x/doc/html/tep123.html.
[38] TinyOS Alliance, TinyOS version 2.x (T2), February 2016, http://www.tinyos.net.
[39] Oracle Labs, SunSPOT World, http://sunspotdev.org/, previously manufactured by Sun Microsystems. (February 2016).
[40] O. Gnawali, TEP 124: The Link Estimation Exchange Protocol (LEEP), 2006, http://www.tinyos.net/tinyos-2.x/doc/txt/tep124.txt.
[41] P. Levis, N. Patel, D. Culler, S. Shenker, Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks, in: Proceedings of the 1st Conference on Symposium on Networked Systems Design and Implementation, 2004, pp. 2–2.

[42] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, P. Levis, Collection Tree Protocol (CTP), in: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, 2009, pp. 1–14.
[43] R. Fonseca, O. Gnawali, K. Jamieson, P. Levis, Four-bit wireless link estimation, in: Proceedings of the Sixth Workshop on Hot Topics in Networks, HotNets VI, 2007, pp. 1–7.
[44] S. Moeller, A. Sridharan, B. Krishnamachari, O. Gnawali, Routing without routes: The backpressure collection protocol, in; Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, 2010, pp. 279–290.
[45] J. Lee, B. Kusy, T. Azim, B. Shihada, P. Levis, Whirlpool routing for mobility, in: Proceedings of the Eleventh ACM International Symposium on Mobile ad hoc Networking and Computing, 2010, pp. 131–140.
[46] M. Richardson, J. Vasseur, IETF Routing Over Low power and Lossy networks (ROLL) Working Group, https://datatracker.ietf.org/wg/roll/charter/, (September 2012).
[47] M. Dohler, T. Watteyne, D. Barthel, Urban wsns routing requirements in low power and lossy networks, Tech. Rep., IETF ROLL, IETF Internet-Draft, 2009.
[48] K. Pister, P. Thubert, S. Dwars, T. Phinney, Industrial Routing Requirements in Low-Power and Lossy Networks, Tech. Rep., IETF ROLL, IETF Internet-Draft, 2009. June.
[49] A. Brandt, J. Buron, G. Porcu, Home Automation Routing Requirements in Low Power and Lossy Networks, Tech. Rep., IETF ROLL, IETF Internet-Draft, 2010. January.
[50] J. Martocci, P. DeMil, W. Vermeylen, N. Riou, Building Automation Routing Requirements in Low Power and Lossy Networks, Tech. Rep., IETF ROLL, IETF Internet-Draft, 2010. January.
[51] T. Winter, P. Thubert, R. Team, Rpl: Ipv6 routing protocol for low power and lossy networks, Tech. Rep., IETF ROLL, IETF Internet-Draft, 2010.
[52] R. Simon Carbajo, M. Huggard, C. McGoldrick, Opportunistic detection of relative mobility in wireless sensor networks, in: Wireless Days (WD), 2010 IFIP, 2010, pp. 1–5.
[53] P. Levis, TOSSIM System Description, Tech. Rep., University of California, Berkeley, 2002.
[54] P. Levis, N. Lee, M. Welsh, D. Culler, TOSSIM: accurate and scalable simulation of entire tinyOS applications, in: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, 2003, pp. 126–137.
[55] H. Lee, A. Cerpa, P. Levis, Improving wireless simulation through noise modeling, in: Proceedings of the 6th International Conference on Information Processing in Sensor Networks, IPSN'07, 2007, pp. 21–30.
[56] J. Banks, J.s. Carson, Introduction to discrete-event simulation, in: Proceedings of the 18th Conference on Winter Simulation, 1986, pp. 17–23.
[57] Atmel, ATmega128L MicroController, February 2016, http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf.
[58] Texas Instruments, CC2420 Chipcon 2.4 GHz Transceiver, http://focus.ti.com/lit/ds/symlink/cc2420.pdf, (February 2016).
[59] Memsic Corporation, MicaZ Datasheet, http://www.memsic.com/wireless-sensor-networks/, previously manufactured by Crossbow Technology Inc. (February 2016).
[60] IEEE, IEEE Std. 802.15.4-2015 - IEEE Approved Draft Standard for Low-Rate Wireless Personal Area Networks (WPANs), January 2015, http://standards.ieee.org/findstds/standard/802.15.4-2015.html.
[61] Memsic Corporation, MIB520CB Datasheet, http://www.memsic.com/wireless-sensor-networks/, previously manufactured by Crossbow Technology Inc., February 2016.
[62] E. Perla, A. O'Cathain, R. Simon Carbajo, M. Huggard, C. Mc GoldRick, PowerTOSSIM z: Realistic energy modelling for wireless sensor network environments, in: 3rd ACM International Workshop on Performance Monitoring, Measurement, and Evaluation of Heterogeneous Wireless and Wired Networks, PM2HW2N 2008, Vancouver, Canada, 2008, pp. 35–42.
[63] V. Shnayder, M. Hempstead, B. Chen, G. Allen, M. Welsh, Simulating the power consumption of large-scale sensor network applications, in: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, 2004, pp. 188–200.
[64] C. Stevens, C. Lyons, R. Hendrych, R. Simon Carbajo, M. Huggard, C. Mc Goldrick, Simulating mobility in WSNs: Bridging the gap between ns-2 and TOSSIM 2.x, in: DS-RT '09: Proceedings of the 2009 13th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications, 2009, pp. 247–250.
[65] M. Zuniga, B. Krishnamachari, Analyzing the transitional region in low power wireless links, in: First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, IEEE SECON 2004., 2004, pp. 517–526.
[66] U. o. S. C. Information Sciences Institute, The network simulator (ns-2), February 2016, http://nsnam.sourceforge.net/wiki/index.php.
[67] Siemens, D6 Wind Turbine Platform - SWT-6.0-154 Brochure and Specifications, February 2016, http://www.energy.siemens.com/hq/pool/hq/power-generation/renewables/wind-power/6_MW_Brochure_Jan.2012.pdf.