Title	Spreading the load in a tree type routing structure
Authors(s)	Delaney, Declan T., Xu, Lina, O'Hare, G. M. P. (Greg M. P.)
Publication date	2013-08-02
Publication information	Delaney, Declan T., Lina Xu, and G. M. P. (Greg M. P.) O’Hare. “Spreading the Load in a Tree Type Routing Structure.” IEEE, August 2, 2013. https://doi.org/10.1109/ICCCN.2013.6614196.
Conference details	2013 22nd International Conference on Computer Communications and Networks (ICCCN), Nassau, Bahamas, 30 July - 2 August, 2013
Publisher	IEEE
Item record/more information	http://hdl.handle.net/10197/7209
Publisher's statement	© © 20xx IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Publisher's version (DOI)	10.1109/ICCCN.2013.6614196

Spreading the Load in a Tree Type Routing Structure

Declan T. Delaney, Lina Xu and Gregory M.P. O'Hare
CLARITY: Centre for Sensor Web Technologies
School of Computer Science and Informatics
University College Dublin
Dublin, Ireland
Email: {declan.delaney, lina.xu, gregory.ohare}@ucd.ie

***Abstract*—Many routing protocols have advanced the Wireless Sensor Network(WSN) paradigm with each new method offering unique ways to maximise Quality of Service(QoS) while minimising energy cost to the network. Tree routing is a well examined method with a proven record in offering a high level of service with trees constructed to fulfil a particular routing objective defined by a chosen metric. The tree structure can be maintained with low overall network overhead but exhibits a weakness with regard to load balancing. Without effective load balancing particular nodes in the network may be subject to excessive network load, leading to uneven energy consumption in the network. This in turn can lead to an unwanted scenario of premature node failure. *Neighbourhood metrics* is presented in this paper as a means to preserve network objectives while achieving improved load distribution in a tree type routing structure. Neighbourhood metrics offer a framework for expanding on currently used metrics to include information on the quality of a nodes *neighbourhood* in addition to the current forwarding route. Neighbourhood metrics is compared to the current state of the art in the form of the Routing Protocol for Low Power and Lossy Networks(RPL) implementation using the *direct* Expected Transmissions(ETX) metric. Neighbourhood metrics exhibits improved load distribution in a number of open public testbeds.**

I. INTRODUCTION

Load balancing in a network is a practice of distributing the communications overhead more evenly over a network. A balanced load is achieved by spreading the sent packets over a greater number of network nodes leaving no node too heavily burdened with packets. Traditionally load balancing is used in traffic engineering, the purpose of which is to improve the end to end latency, throughput or reliability in the network. WSNs on the other hand require load balancing as a means to distribute energy consumption over the network. Each WSN node is reliant on a finite power source making it critical for nodes to minimise power consumption when performing tasks. Powering the radio chip constitutes a large proportion of a nodes overall energy use [1] and each additional packet sent by a node increases its energy spend. Additional energy spend on particular nodes become an issue as these nodes may exhaust power reserves prematurely compromising the network. It is difficult to say at what point a network is compromised as each application and deployment holds specific needs, but it is typical to assume that it is beneficial to maintain activity on all nodes for as long as possible. Although equal load over all nodes is generally neither possible nor desirable it is beneficial to spread the communications load more evenly over the network so individual nodes are not overused.

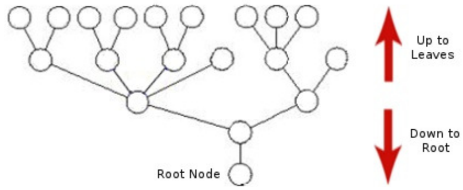

Fig. 1. Routing tree structure

The tree type routing method has become a popular choice for data collection in a WSN. Tree type routing protocols maintain a full routing structure at all times after initialisation. The root node is at the base of the routing tree and acts as the sink for all data collected in the network. Each node in the network maintains a forwarding route to the root node. Figure 1 shows the basic tree structure. Maintaining this structure over the entire lifetime of the network holds a certain communication overhead but does avoid costly route discovery process associated with other dynamic routing protocol [2], [3]. For applications where the collection of data constitutes the majority of the workload in the network a routing tree structure is beneficial. When initialising the routing tree each node chooses the route with the lowest cost metric to the root. Figure 2 illustrates how a metric is propagated throughout a routing tree. The metric here is advertised to all nodes in listening range. Upon hearing the advertisement a link metric is computed and combined with the advertised metric. The node chooses its *parent* as the forwarding node with the best combined score. All traffic through this node will be forwarded to the nodes parent. The node keeps this computed value as its own metric for further advertisement. This process can be described as the *direct metric* method. The *direct metric* value is derived directly from its chosen parent and the link between that parent. The tree develops with the best possible routes to the root as deemed by the metric. Each node will retain information on its immediate neighbours with no information of the wider network available to it. With only local knowledge of the network it is difficult for any node to make an informed

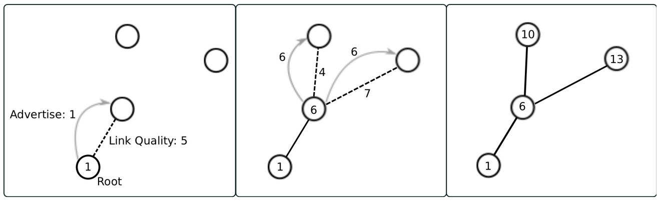

Fig. 2. Metric advertisement in a tree type structure.

decision regarding load balancing. As the routes in a routing tree are defined by the best propagated metric through the network, nodes along the best path to the root can be heavily used with surrounding nodes forwarding all data onto this path leading to a network imbalance.

This paper proposes *neighbourhood metrics*, as a novel framework for combining metrics so a greater distribution of the network load is achieved. Neighbourhood metrics aim to break the dependency on a single best node or path by routing through good neighbourhoods rather than a single good path. A nodes *neighbourhood* is composed of all the nodes within radio range. The neighbourhood metric is a metric composed of all the information available in the neighbourhood. Neighbourhood metrics are an addition to direct metrics where by the advertised metric and link values of all of a nodes neighbours are used to determine the overall metric value of the node, as opposed to direct metrics which use the current parents advertised and link metric only. The neighbour metrics are combined in such a manner so to preserve network stability as discussed in Section III-B. Neighbourhood metrics takes advantage of information already available to a node, negating the need to solicit for additional information off its neighbours. Using a more expansive metric and routing through good neighbourhoods instead of a the single best path allows a greater number of routing options benefiting the network by:

- Promoting a wider spread of network load insuring particular nodes are not used excessively.
- Maintaining the integrity of the current metric keeping on track with network goals.

Section IV presents the benefits of neighbourhood metrics over the current state of the art RPL [4] implementation using direct ETX [5] with implementations using the neighbourhood metrics framework reducing the network load, by up to 35%, over the most heavily burdened nodes.

II. RELATED WORK

Energy awareness is considered a pre-requisite for any WSN protocol. It is then no surprise that all routing methods will maintain some form of power management strategy. Distributing the network load more evenly over the nodes in the network as a means to balance power consumption is considered a paramount priority in a number of popular routing protocol [6], [7], [8], [9]. Each of these protocol use clustering techniques within a hierarchical structure to collect all packets through the network to a data sink. When a cluster is formed, a node is chosen within the cluster as the cluster head. All traffic from the cluster is sent via the cluster head to the data sink. Both the clusters and chosen cluster head change periodically, distributing the network load evenly over the network lifetime. This method is quite effective at balancing the load over the network but requires additional network overhead to determine cluster boundaries and elect a cluster head increasing the overall traffic carried in the network. A Time Division Multiple Access(TDMA) scheme is usually applied over the clusters for access to the cluster head which can result in longer latency on data through the network adding to the cost off applying load balancing in this manner.

Tree type routing protocols were designed in order to achieve high standards on end to end data delivery at low overall energy cost with little consideration made for the distribution of network load. A number of solutions are proposed to support better load balancing in a routing tree. Two common strategies are shown in [10] and [11]. [11] builds a tree structure which maintains effective distribution of load in the network. The method begins building each branch out from the root node. After a node is added to a branch a metric is used to determine how heavily loaded the branch is. The bordering nodes yet to be attached to a branch are then assessed. The heaviest nodes are attached to the branches with the lightest load until all nodes are attached to a branch. The resultant tree provides a well distributed load but at the cost of the quality of the routes chosen. No consideration is given to the quality of the links used in the routes severely effecting the end to end service making this solution difficult to justify for industry adoption. [10] on the other hand does consider the quality of the routes used in the tree. The method uses an established tree and balances the load by allowing nodes to *defect* to other branches if it deems that it will achieve greater load balancing. A number of branch moves can be made in order to achieve a better distribution. The choice to change branch is driven by a heuristic. The paper evaluates three such heuristics with the most effective requiring global knowledge which would not be available in a live deployment. The cost associated with this method is seen by further increasing route instability in the routing structure. Increasing instability is not advised as it can generate additional routing packets in the network or even create short lives routing loops.

Solutions do exist for achieving a more balanced load distribution in a tree routing structure but each come with an associated cost. Reducing the cost of maintaining a balanced load distribution is then a goal for future solutions.

III. NEIGHBOURHOOD METRICS

Neighbourhood metrics provide a framework for metric use. The aim is to use all the information available to the node in order to make better routing decisions. The technique incorporates information regarding the quality of the surrounding neighbourhood into the currently used metric and creates a new metric from which routing decisions can be made. The additional neighbourhood information is combined with the current direct metric value into a single metric that can be used in routing choices. The eventual neighbourhood metric will reflect both the value of the forwarding path and the neighbourhood influence on the node. Maintaining the integrity of the direct metric will make sure any forwarding path through the chosen node is of high quality. Highlighting routes through good neighbourhoods rather than choosing the best single path is advantageous to load distribution in a number of ways:

- Providing a greater number of good, and more closely related neighbourhood alternatives to route through to disincentivise the overuse of highly desirable routing paths.
- Allowing nodes with desirable metric to influence other nodes in its neighbourhood increasing their chances of being chosen as a forwarding route relieving the load.
- Providing a mechanism to maintain stability in the network.

A. Mechanism

The mechanism for providing network neighbourhood metrics is designed to ensure low communication cost is of paramount importance. The process can be broken down into four phases: metric collection, parent selection, neighbourhood metric calculation and metric advertisement.

Metric collection: The first phase requires the collection of metrics used in making routing decisions. The advertisement message's from each of a nodes neighbours are received. Each advertisement message is inspected for both the direct metric and neighbourhood metric. Both metrics may be available from a single advertisement message from a neighbour. The node ID of the neighbouring node, advertising its metric, is also present in the advertisement message. The neighbour ID, advertised direct metric and advertised neighbourhood metric are stored in a *neighbour table*. The node calculates the link metric for each neighbour and stores this metric in the neighbour table. The node updates the neighbour table on a continuous basis as it receives the advertisement messages from its neighbours. When all relevant metrics are collected a node has all the information necessary to select a parent.

Parent selection: Only the neighbourhood metric and link metric are considered when choosing a parent. Combining the link metric and neighbourhood metric will give an overall score for that neighbour. The neighbour with the best overall score is chosen as the nodes parent. The chosen parent is indicated in the neighbour table. After a node has chosen its parent, the neighbourhood metric for the node can be calculated.

Neighbourhood metric calculation: The direct metric and link metric of each of the entries maintained in the neighbour table are used to calculate the nodes neighbourhood metric. An overall score for each neighbour is calculated. The overall score of each neighbour reflects the direct metric the node would assume should it use that neighbour as parent. The neighbourhood metric is determined by combining each score using a weighted function. Scores are weighted depending on their relative distance from the chosen parents score. The weighted function is described in III-B. A node with no neighbours will maintain an undefined neighbourhood metric precluding it from acting as a possible route to other nodes.

Metric advertisement: The final phase entails advertising of metrics to the network. Both the direct metric and the neighbourhood metric must be advertised in order to facilitate the mechanism. The neighbourhood and direct metric may be sent in the same advertisement message. The regulation and timing of advertisement messages is defined by the routing protocol and is out of scope of this paper. The neighbourhood metric for each node is then advertised throughout the network in a similar manner as the direct routing metric until each node has advertised its neighbourhood metric to its neighbours.

Similar to the current practice of using direct metrics, a threshold mechanism is used when a node wishes to switch parent to further increase stability. In order to switch parent, a neighbouring node must advertise a neighbourhood metric that is better than the current parent by, at least, a set limit. This limit is carefully chosen as it poses a trade-off between stability and dynamism; two aspects we wish to maintain in the network.

The strength of this mechanism lies in utilising additional information without necessitating additional communications overhead. The neighbourhood metric is advertised and propagates through the network in the exact same manor as the direct metric advertisements. Each neighbourhood metric, like the direct metric, is composed of a single value. As these messages can be shared, the neighbourhood metric can essentially *piggyback*, causing minimal overhead. The actual cost is the addition of an extra container which will hold the neighbourhood metric of the advertising node. A close link with the direct metric is essential as this will determine the desired behaviour of the network. Advertised neighbourhood metrics are only used for parent decisions and never take part in metric calculations. By using direct metrics during calculations and prioritising the chosen parent value in these calculations a strong link is maintained between the neighbourhood metric and the direct metric.

B. Weighting neighbours

The means by which neighbourhood metrics are calculated is pivotal in achieving the desired behaviour from the network. The neighbourhood metric should reflect the real quality of a given route chosen but also carry information regarding the quality of the surrounding nodes. This must be achieved using a single value metric. A single value metric removes any ambiguity for a node when choosing a parent. If a neighbour in the neighbourhood has the single best neighbourhood metric value, this becomes the preferred parent with which to forward data. If two neighbours exhibit the same, best value, a simple choice for preferred parent can be made using the node ID. The neighbourhood metric for a node is calculated by combining the nodes direct metric value with a value derived from the direct metric of each, or a subset, of its neighbours. This is accomplished by starting with the direct metric and adding a neighbourhood effect from each of its neighbours. How each neighbour effects the overall neighbourhood metric is described by a set of directives which are defined in this section.

Use of direct metric: The direct metric used in the network will determine the real quality of the route used. The real quality is important as it produces the response desired from the network. It is important for the neighbourhood metric to reflect the real quality of the route, thus the direct metric for the current forwarding route plays a strong role while calculating the neighbourhood metric. In fact the direct metric must remain the primary component of the resulting metric in order to maintain the characteristics of the direct metric that provide the desired requirements to the network.

Positive effect: To further define the nature of the neighbourhood metric, examination of a scenario is helpful. Figure 3 highlights a scenario where a node can choose a routing path between two neighbours. The figure shows a network of nodes with data source, and a data sink marked as root. The source must choose a path to the root through either neighbour **A** or **B**. Neighbour **A** has two neighbours, one with good links to the root, which result in a good direct metric. One with bad links to the root, resulting in a poor direct metric value. Neighbour **B** has a single neighbour with good links to the root. Despite the poor quality of the additional route which **A** provides, it is reasonable to assume route **A** still provides a

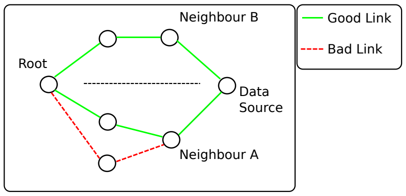

Fig. 3. Choosing a parent using neighbourhood knowledge.

marginal advantage over route **B**. Each additional neighbour will then have a positive effect on the neighbourhood metric. This is not to say that it is worth *changing* parent from node **B** to node **A** if this scenario occurs. A stability bound as discussed in section III-A is important in this regard. The stability bound will determine at what point the node should change parent over to an improved route. A change in route should only occur if there are significant improvements to the real route value. The value of the neighbourhood metric must minimise influence to promote a parent change but must still influence a nodes ability to be chosen. The neighbourhood effect applied to a node by its neighbours will then be relative to the stability bound. The neighbourhood effect on the metric must never surpass the stability bound resulting in a parent switch, maintaining stability in the network.

Diminishing return: Each neighbouring node, which is part of the routing tree, will offer a route through which the node can send information. Each neighbour, not chosen as parent, can be considered to offer a failover route should the current route become unavailable or compromised in some way. Each neighbour will increase a nodes probability of maintaining a route to the root. Each additional neighbour that is added however delivers a diminishing return on maintaining a route as it represents probabilities in parallel. The neighbour, which is not the parent, that offers the best routing option adds the largest effect to the metric as it is most likely to be chosen if the parent becomes unavailable. This effect must be given consideration when calculating the neighbourhood metric.

Similar routing options: In order to reduce major fluctuations in routing metric when a disturbance occurs that requires a parent change, a node should change to a parent that presents a similar routing metric to the original route before the disturbance. The node will then maintain and advertise a similar routing metric which will not affect the rest of the branch greatly. This concept is key to maintaining stability in the routing tree. Routing through nodes which have multiple routes of similar quality is therefore advisable. This effect will also be considered when calculating neighbourhood metric.

From these observations a set of directives can be used which will promote routing stability when creating the neighbourhood metric for a given node:

- The *direct metric* of the chosen parent must remain the primary component of the resulting neighbourhood metric.
- Each neighbour ought to have a *positive effect* on the neighbourhood metric, with the effect considered as a fraction of the stability bound.
- Each additional neighbour has a *diminishing return* on the neighbourhood metric.

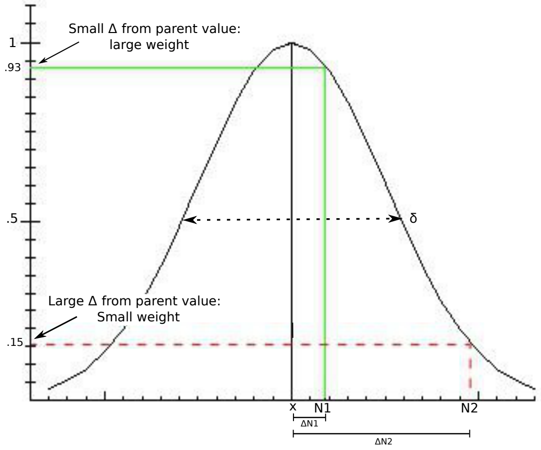

Fig. 4. Using a Gaussian weighting function to weight the neighbourhood effect.

- The neighbourhood metric should promote nodes which have many *similar routing options*. The neighbour effect of any neighbour will be greater as the value tends towards the chosen parent metric value.

Figure 4 shows how a Gaussian weighting function can be used in weighting neighbour metrics for use in calculations. The Gaussian function maintains a value of 1 at its peak. A node will use this function to give a weight to each of its neighbours. The direct metric value of each neighbour and the node its self are plotted on the x-axis. The Gaussian curve is centred on the nodes own direct metric value. Centring the expected value of the curve on the parent metric value promotes neighbours with a similar metric while penalising neighbours of dissimilar metric. The Gaussian fall off graduation will determine how heavily each node is weighted. The figures show the weighting applied to two neighbours: N1 and N2. N1 is considered a good neighbour and has a similar metric value to the chosen parents value. N2 is a bad neighbour as it has a metric far removed from the parent metric. By applying the Gaussian weighting function we generate a heavy weight to the good neighbour and lighter weights to the bad neighbours.

We can define and implement these directives in a simple equation. Equation 1 shows the equation used in calculating the neighbourhood effect of a single neighbour on the node.

$$\text{Neighbourhood Effect} = e^{-\frac{\Delta v^2}{2\times\delta^2}} \times \frac{1}{i^2} \times \Theta \tag{1}$$

Algorithm 1 describes how the combination of this effect produces the neighbourhood metric for a node. $e^{-\frac{\Delta v^2}{2\times\delta^2}}$ determines the weighting applied to the neighbour as a function of its value and how close this value lies to the parent metric value, where Δv is the difference between the current parents value and the neighbours value and δ is the Gaussian variance used to control the width of the function. $\frac{1}{i^2}$ provides a diminishing return in weight to each additional neighbour as a quadratic falloff, with i the index of the neighbour at hand. Θ is the stability bound for the metric.

The equations above are used in a simple algorithm to calculate each neighbours neighbourhood effect and the resulting neighbourhood metric for the node. The algorithm first sorts the list of neighbours into highest value to lowest. The neighbourhood effect of each neighbour is then calculated. The effect of each neighbour is accumulated and then added or subtracted to the nodes direct metric to make the nodes neighbourhood metric. Using this method the direct metric of the parent remains the most prominent factor in the neighbourhood metric, each additional neighbour provides a diminishing return with a quadratic falloff effect. A node initially holds an "infinite" value for its metric precluding it from neighbour calculations or acting as a forwarding route. Only when a node hears advertisements from neighbouring nodes can it calculate its own metric and join the routing tree.

```
$neighbour\_set \rightarrow$ List in order of descending value;
$neighbourhood\_effect \leftarrow 0$;
for $i = 1 \rightarrow neighbour\_set.length$ do
    $diff = node.value - neighbour[i].value$;
    $neighbourhood\_effect += e^{-\frac{\Delta v^2}{2 \times \delta^2}} \times \frac{1}{i^2} \times \Theta$;
end
if min metric desired then
    $neighbourhood\_metric \leftarrow$
    $direct\_metric - neighbourhood\_effect$;
else if max metric desired then
    $neighbourhood\_metric \leftarrow$
    $direct\_metric + neighbourhood\_effect$;
```

Algorithm 1: Calculating Neighbourhood Metric for a Node

C. Weaknesses

The neighbourhood metric technique makes use of information that is readily available to the node but does require an amount of overhead for its use. Additional overhead associated with running neighbourhood metrics are incurred by increasing algorithmic complexity and increase in ram usage with extended neighbour table. Increasing algorithmic complexity incurs additional power usage in processing. The algorithm shows linear complexity and is performed over the neighbour table which can be considered a small data set.

Distorting the direct metric can be viewed as a negative move as the node no longer sees the absolute best path available to it as defined by the metric. The nodes can then choose routes that are not considered the best choice to illicit the required response from the network. This effect however, is necessary in order to achieve better distribution over the network. As the neighbourhood metric is driven by the original metric the neighbourhood metric maintains a close relationship to the desired response from the network. Therefore a tradeoff exists between choosing the single best path to deliver data and choosing good quality paths with greater load distribution in the network.

IV. EXPERIMENTATION AND RESULTS

A number of experiments are completed in order to validate the neighbourhood metric method. The goal of the experiments was to compare the neighbourhood metric method with the current state of the art and determine how well each solution distributes network load through the network. The number of packets forwarded by each node was monitored during experimentation to determine network load. An application based on a Building Energy Management System(BEMS) was used during experiments. BEMSs constitute a realisable industry application for WSN and have contributed to the design and direction in which WSNs can focus to achieve adoption as a viable solution in industry.

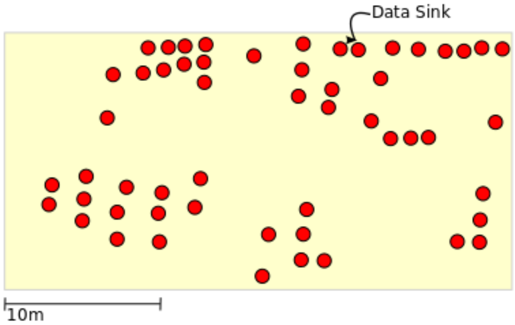

Fig. 5. Node orientation in Wisebed testbed

TABLE I. SCALE AND DENSITY OF THE WISEBED AND KANSEI GENIE ENVIRONMENTS.

	Wisebed	Kansei Genie
Number	49	298
Scale	1.82	2.84
Density	23.5	38.8

***Scale:* Average number of hops to the root.**
***Density:* Average number of neighbours to a node.**
***Number:*The amount of nodes used, not including damaged/unavailable nodes**

A. Experimental Environment

Experiments were conducted over two public experimental testbeds. The Wisebed testbed [12] situated in the University of Lübeck, Germany and the Kansei Genie testbed [13] situated in the University of Ohio, Ohio, USA were used for experiments. The Wisebed testbed facilitates access to 54 nodes in a network deployed in a working office environment. Figure 5 shows the orientation of the nodes within the building. Using this testbed gives an insight into how each solution copes in a real dynamic office environment, remaining in line with the BEMS application scenario. The Kansei Genie testbed contains up to 360 nodes in a grid structure in a purposely built environment. This will give insight into both solutions at higher scale and density. Figure 6 shows the node orientation for this testbed. The two experimental environments used differ with regard to scale, density and physical dynamics allowing the results to be generalisable over a large range of deployment types. The Wisebed testbed is situated in a working office so many physical events will occur within and around the nodes communication channels providing a more dynamic radio environment. Table I presents the average scale and density of both testbeds.

B. Experimentation

All experiments were conducted using TelosB nodes [14]. TinyOS [15] was used as the platform for conducting experiments. All nodes contributing to the experiment run the

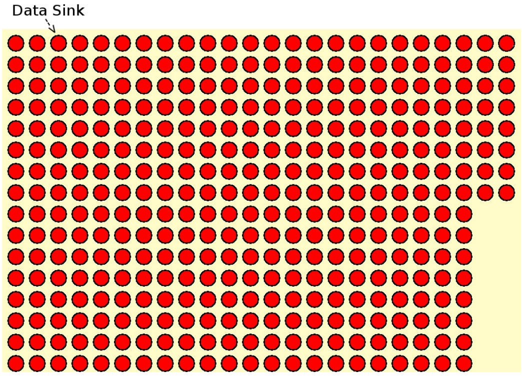

Fig. 6. Node orientation in Kansei Genie testbed

TinyOS architecture. The RPL routing protocol was used as the routing protocol for all experiments. RPL is fast becoming a new standard for WSN protocol and represents the state of the art for industry deployment. The standard "out of the box" RPL implementation using the "minimum rank with hysteresis" object function [16] was used representing RPL using Direct Metrics(RPL-DM). A RPL implementation using the ETX metric within the neighbourhood metrics framework was used representing RPL using Neighbourhood Metrics(RPL-NM). The evaluation was conducted to compare the load distribution capabilities of both RPL-DM and RPL-NM. Both implementations use the ETX link metric for link estimation. ETX is an extensively tested metric providing reliable quality link estimation. The experiments are conducted over one hour intervals, where the network is set up and maintained for this hour. The experimental scenario considered reflects a simple BEMS application, where each node senses the environment and sends this information to a collection point. A focus is given to two main data flow types:

- **Periodic:**A packet is sent from each node at regular intervals. The frequency of packets varies with a node choosing a random wait period between 30 and 90 seconds. This represents a fine grained monitoring system for a BEMS dealing with building environmental control.
- **Bursty:**A number of packets are sent in quick succession, a burst, with longer periods between bursts. The packet frequency during a burst is 1 per second and the wait period between bursts is selected at random between 30 and 90 seconds. This can represent an emergency event or numerous high priority events occurring simultaneously.

C. Results

The benefits of distributing the network load are more keenly observed on the most heavily burdened nodes. These nodes carry the majority of the packets through the network and are at risk of premature failure due to energy overuse. Figure 7 shows a comparison between RPL-DM and RPL-NM carried out on the Wisebed testbed. The figure shows the the load carried in the ten most heavily burdened nodes in the network as a percentage of the overall load carried in the whole network. "1" indicates the percentage carried by the most heavily burdened node, "2" indicates the percentage load carried by the two most heavily burdened nodes with "10" indicating the percentage load carried by all of the ten most heavily burdened nodes. Naturally an increasing trend is observed as each additional node will increase the overall load on the node set. Using direct metrics the most heavily burdened node carries almost 50% of the overall load. This is reduced significantly to 30% using neighbourhood metrics. This trend is carried throughout the data with neighbourhood metrics easing the load on the most heavily burdened nodes. The most heavily burdened nodes carry a large percentage of the overall load in the Wisebed testbed owing the the small scale and even perhaps the particular orientation of the nodes. These figures are reduced in a larger scale network which offers many more alternative routing paths. Figure 8 shows a similar graph applying both RPL-DM and RPL-NM to the Kansei Genie testbed. The overall load on the most heavily burdened nodes are reduced as expected. RPL-NM still presents a clear advantage with regard to load distribution over the nodes with noticeable reductions of 10-15% on load over the nodes

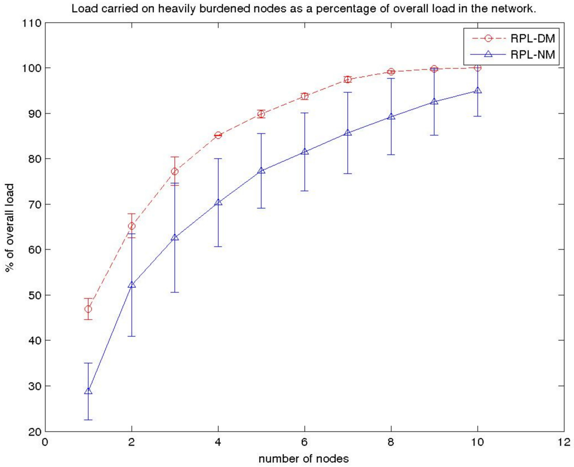

Fig. 7. Load distribution over heavily burdened nodes under periodic traffic on the Wisebed testbed

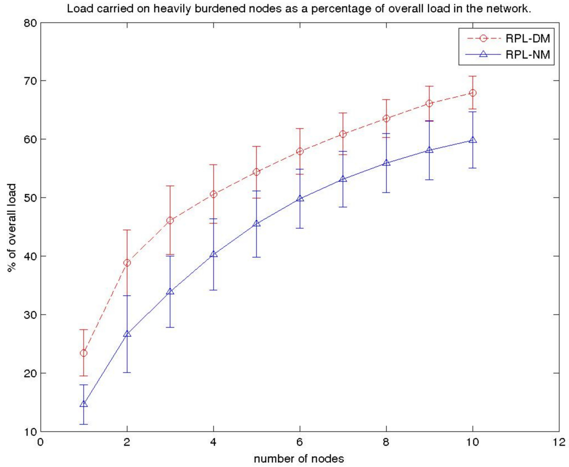

Fig. 8. Load distribution over heavily burdened nodes under periodic traffic on the Kansei Genie testbed

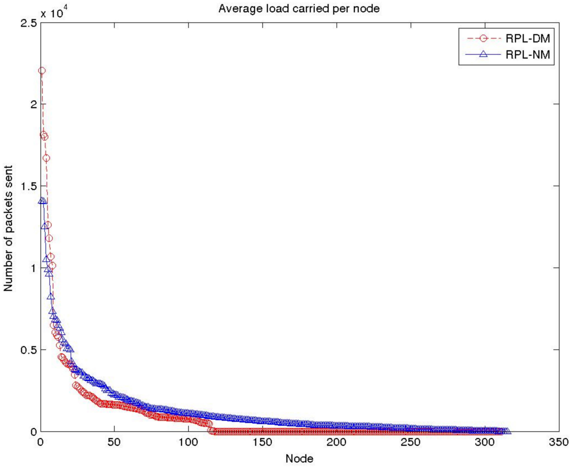

Fig. 9. Load carried per node under bursty traffic in the Kansei Genie testbed

shown. The bursty traffic scenario leads to a very congested network. In this scenario the perceived good routes are even more heavily used and get congested. This can degrade the performance of the route leading to route changes and hence a higher percentage of nodes carrying traffic is noticeable. For periodic traffic, 29% of nodes carry significant traffic through the network using RPL-DM on the Kansei Genie testbed. This increases to 38% using bursty traffic. Using RPL-NM with bursty type traffic on the Kansei Genie testbed, 86% of nodes carry significant traffic through the network. Figure 9 depicts the average number of packets carried by each node in the network under bursty traffic and shows this phenomenon quite well. This effect is less profound in the Wisebed testbed with 28% of nodes carrying traffic using RPL-DM and 31% of nodes carrying traffic using RPL-NM owing to the scale of the network and reduced levels of congestion.

V. CONCLUSION AND FUTURE WORK

Energy efficiency and improved QoS are two main goals for any WSN routing protocol. Tree based routing structures are designed to achieve both low overall energy consumption and quality routes but do however exhibit a frailty with regard to balancing the energy use over the network. This leaves some nodes exposed to energy overuse and premature failure. In this paper we introduce neighbourhood metrics, a novel framework for use with metrics in a routing tree structure which allows identification of good routing neighbourhoods instead of a single best routing path. Using neighbourhood metrics promotes the use of a more diverse set of forwarding routes, allowing greater distribution of the network load, without compromising the integrity of the underlying goal of the network. This is achieved by exploiting additional information already available to the node and does not necessitate additional overhead in the network. Results show a reduction of load of up to 35% on heavily burdened nodes in the network using neighbourhood metrics over the current state of the art. Currently the neighbourhood metrics framework is implemented for the RPL protocol using the ETX link metric. Expanding the scope of this work with implementation on other tree type protocol and a wider set of metrics is necessary and is the next logical step for future work.

ACKNOWLEDGMENT

This work is supported by Science Foundation Ireland under grant 07/CE/I1147. The authors would like to thank Daniel Bimschas at the University of Lübeck and Michael McGrath at the University of Ohio for handling any and all queries with regard to the Wisebed and Kansei Genie testbeds, answering the whats and whys with patience and promptly dealing with any issues at hand.

REFERENCES

[1] O. Landsiedel, K. Wehrle, and S. Gotz, "Accurate prediction of power consumption in sensor networks," in *Proceedings of the 2nd IEEE workshop on Embedded Networked Sensors*, ser. EmNets '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 37–44.

[2] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," in *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, feb 1999, pp. 90 –100.

[3] C. Intanagonwiwat, R. Govindan, and 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*, ser. MobiCom '00. New York, NY, USA: ACM, 2000, pp. 56–67.

[4] Winter and et al, "Rpl: Ipv6 routing protocol for low power and lossy networks," IETF-ROLL, Tech. Rep., 2012.

[5] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proceedings of the 9th annual international conference on Mobile computing and networking*, ser. MobiCom '03. New York, NY, USA: ACM, 2003, pp. 134–146.

[6] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, jan. 2000, p. 10 pp. vol.2.

[7] A. Manjeshwar and D. Agrawal, "Teen: a routing protocol for enhanced efficiency in wireless sensor networks," in *Parallel and Distributed Processing Symposium., Proceedings 15th International*, april 2001, pp. 2009 –2015.

[8] O. Younis and S. Fahmy, "Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 3, no. 4, pp. 366 – 379, oct.-dec. 2004.

[9] S. Bandyopadhyay and E. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3, march-3 april 2003, pp. 1713 – 1723 vol.3.

[10] P.-H. Hsiao, A. Hwang, H. Kung, and D. Vlah, "Load-balancing routing for wireless access networks," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, 2001, pp. 986 –995 vol.2.

[11] H. Dai and R. Han, "A node-centric load balancing algorithm for wireless sensor networks," in *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, vol. 1, dec. 2003, pp. 548 – 552 Vol.1.

[12] I. Chatzigiannakis, C. Koninis, G. Mylonas, S. Fischer, and D. Pfisterer, "WISEBED: an open large-scale wireless sensor network testbed," in *Proceedings of the 1st International Conference on Sensor Networks Applications, Experimentation and Logistics*, Sep. 2009.

[13] E. Ertin, A. Arora, R. Ramnath, M. Nesterenko, V. Naik, I. Bapat, V. Kulathumani, M. Sridharan, H. Zhang, and H. Cao, "Kansei: A testbed for sensing at scale," in *in Proceedings of the 4th Symposium on Information Processing in Sensor Networks (IPSN/SPOTS track*. ACM Press, 2006, pp. 399–406.

[14] "XBOW sensor motes specifications," 2008, http://www.xbow.com.

[15] P. Levis, S. Madden, J. Polastre, R. Szewczyk, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, "Tinyos: An operating system for sensor networks," in *in Ambient Intelligence*. Springer Verlag, 2004.

[16] O. Gnawali and P. Levis, "The minimum rank with hysteresis objective function," IETF-ROLL, Tech. Rep., 2012.