A

Neighbour-Disjoint Multipath for Low-Power and Lossy Networks

A.K.M. MAHTAB HOSSAIN, University of Greenwich
CORMAC J. SREENAN, University College Cork
RODOLFO DE PAZ ALBEROLA, United Technologies Research Center (UTRC) Ireland Ltd.

In this paper, we describe a neighbour disjoint multipath (NDM) scheme that is shown to be more resilient amidst node or link failures compared to the two well-known node disjoint and edge disjoint multipath techniques. A centralised NDM was first conceptualised in our initial published work utilising the spatial diversity among multiple paths to ensure robustness against localised poor channel quality or node failures. Here, we further introduce a distributed version of our NDM algorithm adapting to the low-power and lossy network (LLN) characteristics. We implement our distributed NDM algorithm in Contiki OS on top of LOADng – a lightweight On-demand Ad hoc Distance Vector Routing protocol. We compare this implementation's performance with RPL – a standard IPv6 routing protocol of LLN, and also with basic LOADng, running in the Cooja simulator. Standard performance metrics such as packet delivery ratio, end-to-end latency, overhead and average routing table size are identified for the comparison. The results and observations are provided considering a few different application traffic patterns, which serve to quantify the improvements in robustness arising from NDM. The results are confirmed by experiments using a public sensor network testbed with over 100 nodes.

CCS Concepts: •**Networks** → **Network algorithms;** *Routing protocols;* •**Computer systems organization** → **Sensor networks;** *Redundancy;*

Additional Key Words and Phrases: Neighbour disjoint multipath (NDM), wireless sensor networks, node-disjoint multipath, edge-disjoint multipath, LOADng, RPL

1. INTRODUCTION

With the advent of sensor technology, a new class of multi-hop wireless sensor network (WSN) emerged which is generally characterised by a resource-constrained failure-prone architecture, and subsequently has given rise to new challenges in order to provide robustness or resilience. Among many applications, these types of WSN are used in surveillance, natural disaster monitoring, and for industrial process monitoring and control where a certain reliability should be ensured while providing robustness in the presence of harsh surroundings [Sha et al. 2013; Radi et al. 2012].

The lossy links and low cost, low power nature of WSNs challenge the use of these sensors in applications that demand high reliability and low latency. The harsh wireless environment results in failures in WSNs that may be classified into i) transient failures, or ii) permanent failures [Avizienis et al. 2004]. Transient failures usually affect communication links between sensors, and may be caused by interference, multipath fading, and other environmental factors. Sometimes these failures can be long-lived if an interference source is persistent. Permanent failures are due to hardware

A preliminary version of our work on NDM was published in IEEE DCOSS 2014 [Hossain et al. 2014].
Author's addresses: A.K.M. Mahtab Hossain is with the Department of Computing and Information Systems, University of Greenwich (E-mail:a.k.m.hossain@gre.ac.uk); this research was completed while he was at University College Cork. Cormac J. Sreenan is with the Department of Computer Science, University College Cork (E-mail: cjs@cs.ucc.ie). Rodolfo de Paz Alberola was with the United Technologies Research Center (UTRC) Ireland Ltd. (Email: DePazAR@utrc.utc.com) when this work was completed.

DOI: http://dx.doi.org/10.1145/0000000.0000000

malfunction, node destruction, or energy depletion of sensors. Both types of failures can be detrimental for a WSN application having stringent reliability requirements, since they might cause system failures which can result in economic loss, environmental damage or other serious consequences. While the literature has focused on failures that affect nodes/links in an isolated or random manner, in reality it would be expected that some failures, such as those caused by radio interference or node destruction, would impact on multiple links and nodes that are located nearby to each other. This observation provided the motivation for us to devise a multipath technique for WSNs that utilises radio-disjointedness as the key property in terms of neighbour selection for multiple routing paths.

In this article, we describe a neighbour disjoint multipath (NDM) scheme that is shown to perform better in failure-prone scenarios than the two well-known node disjoint (NODE) and edge disjoint (EDGE) multipath techniques. Two paths are edge or node disjoint if there have no edge or node in common, respectively [Cormen et al. 2001]. NDM is conceptualised in order to minimise the impact of co-located node or link failures. Such phenomenon is observed during interference or jamming or large scale damage where a localised portion of the network might be unusable. MAC protocol solution and retries are not enough for such scenarios as they are kept quite simplistic for WSN. The NDM scheme chooses the shortest path between a sensor and the sink as the primary path. Then it tries to select a set of backup paths that have no node that belongs to the primary path or even a neighbour to any node of it except source or sink. In selecting the backup paths, we utilise the disjoint property to ensure that i) when there are k paths between source and sink, no set of k node failures can result in total communication break between them, and ii) by having $(k-1)$ spatially separated backup paths w.r.t. the primary path, the probability of simultaneous failure of the primary and backup paths is reduced in case of localised poor channel quality or node failures.

A centralised NDM was first conceptualised in our previous published work [Hossain et al. 2014]. In this article, we prove its optimality, and further introduce a distributed version of our NDM algorithm adapting to the low-power and lossy WSN characteristics. We implement it in Contiki OS on top of LOADng [Clausen et al. 2014] – a lightweight On-demand Ad hoc Distance Vector Routing protocol. We compare this implementation's performance with RPL [Winter et al. 2012] – a standard IPv6 routing protocol of LLN, and also with basic LOADng, running in the Cooja simulator. In the literature, RPL and LOADng are often compared with respect to (w.r.t.) various LLN specific performance metrics, and application types [Yi et al. 2013; Herberg and Clausen 2011; Vučinić et al. 2013]. This also motivated us to select these two well-known protocols for comparison. Standard performance metrics such as packet delivery ratio, end-to-end latency, overhead and average routing table size are identified for the comparison. The results and observations are provided considering a few different application traffic patterns, which serve to quantify the improvements in robustness arising from NDM. The results are confirmed by experiments using a public sensor network testbed with over 100 nodes. NDM was seen to improve basic LOADng's performance in terms of reliability and latency while slightly increasing its overhead.

The rest of the paper is organised as follows. We discuss related work in Section 2. We provide a review of our centralised NDM algorithm, and also present some newer findings in Section 3. The distributed NDM algorithm, and its implementation are discussed in detail in Section 4. In Section 5, we present both simulation and real-world experimental findings, and point out the effectiveness of our NDM approach compared to two well-known LLN routing protocols, namely, RPL and LOADng. Finally, we depict in Section 6 the conclusions drawn, and our future work.

2. RELATED WORK

NDM routing was conceptualised in order to ensure resilience against node or link failures. Different approaches could be undertaken to address the same issue, such as i) retransmission based strategy where both the sender and receiver might instigate the retransmission, and ii) by introducing redundancy in the form of antenna/node duplication [Willig 2005; Sitanayah et al. 2014], sending the same information multiple times [Annamalai and Bhargava 1998] or incorporating error correction codes inside the packet, etc. There are many routing protocols in WSN developed over the years. Almost all the WSN routing protocols can largely be divided into three main categories: i) *data-centric* which differs from the traditional address-centric routing in a way that data from multiple sensors can be aggregated, and thereby reducing the number of redundant transmissions towards the sink [Kulik et al. 2002; Intanagonwiwat et al. 2003], ii) *hierarchical* in terms of clustering of nodes where the cluster head performs some aggregation and reduction of data for energy conservation [Heinzelman et al. 2002; Lindsey and Raghavendra 2002] or iii) *location-based* that utilises position information to route data to the desired regions [Yu et al. 2001; Nath and Niculescu 2003]. Apart from the usual performance metrics of a routing protocol (e.g., packet delivery, latency, control message overhead, etc.), resource constraint (i.e., energy and memory) of the tiny sensors plays a vital role in the design of these protocols. More in-depth discussion can be found in [Akyildiz et al. 2002; Patil and Biradar 2012]. In addition, there are a few distinct ones based on network flow or quality of service (QoS) awareness (e.g., SAR [Sohrabi et al. 2000], SPEED [He et al. 2003], etc.), and multipath-based protocols where our contribution lies.

The literature on multipath routing in WSN is also vast, and we do not aim to be comprehensive in our survey. A recent detailed survey on multipath techniques in WSN could be found in [Sha et al. 2013; Radi et al. 2012]. We briefly outline some research that utilise spatial diversity for finding multiple paths similar to us. Maximally radio-disjoint multipath routing (MR2) [Maimour 2008] adopts an incremental approach to construct the minimum interfering paths in satisfying an application's bandwidth requirements. [Tsai and Moors 2007] proposes a weighted interference multipath metric that takes into account the spatial diversity through introducing the number of neighbours in estimating interference. The higher the number of neighbours of a node the higher its interference multipath metric, which gives a notion of the node's interference encountered. Wu and Harms [Wu and Harms 2001] try to select least-correlated paths using the number of link connectivity among the paths. Interference minimised multipath routing (I2MR) tries to construct zone-disjoint paths with the requirement of localisation support inside the sensors [Teo et al. 2008]. A few others also try to achieve zone-disjointedness using directional antennae [Roy et al. 2002]. However, all these zone-disjointed schemes require special hardware/service related to localisation inside the resource constrained sensors.

All these papers try to adopt spatial diversity in order to find multiple paths like us. However, they are designed with specific application in mind. They are protocol driven scheme with the goal of achieving a specific application's requirement, e.g., high-speed multimedia streaming, higher throughput [Maimour 2008], etc. or require specialised hardware support [Teo et al. 2008; Roy et al. 2002]. Our work is more fundamental in the sense that could be utilised in a setting just like the basic node disjoint or edge disjoint algorithms where no node or edge is shared among the primary and backup paths, respectively. We will refer to node disjoint and edge disjoint multipath techniques as NODE and EDGE, respectively from here on.

Since NDM aims to provide resilience, it is relevant for various applications that are characterised by Low Power and Lossy Networks (LLNs) including Wireless Personal

Area Networks (Spans), integrated low-power Power Line Communication (PLC) networks and WSNs. NDM could work with any routing protocol where its main job is to find alternative paths w.r.t. the main routing path. We implemented a distributed NDM on top of LOADng (Lightweight On-demand Ad hoc Distance-vector routing protocol - Next Generation) [Clausen et al. 2014]. LOADng is one of the routing protocols proposed to work with IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) standard [Montenegro et al. 2007] mandated by Internet Engineering Task Force (IETF). RPL is the standard routing protocol designed by IETF ROLL working group to tackle the issues inherent to LLNs [Winter et al. 2012]. Since we compare our implementation with both RPL and LOADng, we provide a brief description of both the protocols in the following.

RPL is a proactive distance vector routing protocol that is meant to support multipoint-to-point (MP2P) (sensor to controller/sink) and point-to-multipoint (P2MP) (controller/sink to sensor) routing in an efficient way. It operates by creating a destination oriented directed acyclic graph (DODAG) rooted at sink/controller by optimising some link reliability metric, such as expected transmission count (ETX), hop count, link latency, throughput, node energy or even some user defined metric. The above metrics can also be used as constraints, e.g., as a rule specifying not to include the nodes inside the routing path having remaining energy less than a threshold. It supports peer-to-peer (P2P) conventional routing too but in a complicated way as all such P2P communication paths may go through the DODAG root resulting in sub-optimal routes. RPL also offers redundant paths between a sensor and the sink, however, no notion of disjointedness is applied for their construction. Since it is a proactive protocol maintaining fresh routes from sink to sensor and vice versa all the time, it is expected to provide superior latency performance but generates increased overhead [Winter et al. 2012]. The DODAG is constructed through the transmission of DIO (DODAG Information Object) messages. The downward routes from the sink towards the sensors are setup by the Destination Advertisement Object (DAO) control messages. The trickle algorithm [Winter et al. 2012] governs the emission interval of the control messages. It works with the principle of reducing the control overhead by sending them less frequently when there is no change in the topology, and more frequently when changes are detected.

On the other hand, LOADng is a reactive routing protocol in order to eliminate unnecessary traffic overhead of RPL with the assumption that LLNs are idle most of the time. It is a lightweight version of mobile ad-hoc network (MANET) routing protocol, AODV [Perkins and Royer 1999] adapted for LLNs that only searches for routes on demand. It works according to the same principle as AODV using Route-Request (RREQ) and Route-Reply (RREP) messages to setup reverse and forward P2P routes between source and destination, respectively [Clausen et al. 2014]. LOADng is claimed to reduce the overhead generation when there are not a large number of P2P traffic active at the same time. It increases the packet delivery latency though because of its initial route discovery phase.

3. CENTRALISED NDM

The Neighbour Disjoint Multipath (NDM) constructs a primary path, and a set of alternative or backup paths between a source and the sink. It strives to achieve a set of backup paths that are neighbour-disjoint w.r.t primary path. In other words, they have no node that is inside the primary path or even neighbour to any node of it. The primary path between a source and the sink is generally the shortest path between them [Sha et al. 2013; Radi et al. 2012]. We also retain this concept of primary path in the design of NDM scheme. In exploring the alternative or backup paths, we try to exploit spatial placement diversity among the nodes: i) no node except the source and

sink of these paths is part of the primary path, and ii) any node except the source and sink of these paths is preferably not a neighbour to any other nodes of the primary path. Because of the design principles used, these backup paths are expected to be unaffected by localised path failures, i.e., simultaneous destruction of sensors confined to a specific area or correlated bad channel conditions.

The rest of the section is organised as follows. First, we provide the formal definitions and preliminaries required to explain the NDM algorithm. We then discuss the centralised NDM algorithm in detail with an illustrative example in Section 3.2. We also prove the algorithm's optimality in terms of finding the least correlated backup paths. We list the performance metrics that have been used in this article for evaluation purpose in Section 3.3. We then provide a brief comparison of centralised NDM with other disjoint multipath schemes in Section 3.4.

3.1. Definitions and Preliminaries

Suppose a WSN topology is represented by an undirected graph, $G = (V, E)$ where V and E represent the set of sensor nodes, and the set of edges among them. An edge $(u, v) \in E$ indicates sensor nodes u and v can communicate with each other. A path between source, s and destination t is a sequence of vertices along the path, $P_i =< v_{i1}, v_{i2}, \ldots, v_{iM} >$ where $v_{i1} = s,\ v_{iM} = t,$ and $(v_{im}, v_{i(m+1)}) \in E$, $m = 1, 2, \ldots, (M-1)$. Primary path, PP is the shortest path between s and t, i.e., $|PP| \leq \forall_i |P_i|$.

Assume V_i denotes the set of vertices of the primary path, and $\mathcal{N}(v_{im})$ denotes the neighbour set of vertex $v_{im} \in V_i$. A pure NDM backup path is defined as $P_j =< v_{j1}, v_{j2}, \ldots, v_{jL} >$ where $v_{j1} = s,\ v_{jL} = t$, and the following conditions are satisfied,

$$v_{jl} \notin V_i - \{s, t\}, \tag{1}$$

$$v_{jl} \notin \forall_{m, m \neq s,t} \mathcal{N}(v_{im}). \tag{2}$$

where $l = 2, 3, \ldots, L-1$. Condition (1) implies that no primary path node other than the source and destination will be inside the backup path. Condition (2) states that except for the source and destination, no node inside the backup path will be a neighbour of any primary path's node. However, there might be some backup paths where (2) is not satisfied. For such impure NDM backup paths, a non-negative weight function w is defined where $w(P_j) = \sum_{l=1}^{L} \rho(v_{jl})$ denotes the sum of the correlation weights of the vertices along the backup path P_j. The correlation weight $\rho(v_{jl})$ of vertex v_{jl} is defined as,

$$\rho(v_{jl}) = \begin{cases} 1 & \text{if } v_{jl} \notin \{s, t\} \ \&\ v_{jl} \in \exists_{m, m \neq s,t} \mathcal{N}(v_{im}) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

We define the least correlated NDM backup path (BP) weight, $\delta\rho(s, t)$ from s to t by,

$$\delta\rho(s, t) = \begin{cases} \min\{w(P_j)\} & \text{if a backup path exists} \\ \infty & \text{otherwise} \end{cases}$$

3.2. NDM Construction Algorithm

The centralised NDM algorithm assumes global knowledge of the network topology in terms of neighbour information, and works in the following manner:

— First, the primary path which is the shortest path between source and sink is computed.
— Then a set of backup paths that are neighbour-disjoint w.r.t. primary path is constructed incrementally until no other such path exists. The backup paths may consist of two different types of NDM: i) pure ($w(P_j) = 0$) and ii) impure ($w(P_j) > 0$).

ALGORITHM 1: NDM (G, s, t, w, ρ, K)

s = source, t = sink, and K = number of paths computed so far. The *colour* variable associated with each vertex indicates if it is already a part of the previously computed paths (BLACK), inside the queue (GREY) or unexplored (WHITE). Weight e is obtained from topology information where $e(u, v) = 1$ if $(u, v) \in E$ and $e(u, v) = 0$ otherwise. Correlation factor associated with each vertex $v \in V$ is defined in Eq. (3). $Adj[u]$ denotes neighbours of u.

```
1: INITIALISE (G, s, t, ρ, K)
2: Q ← {s}
3: colour[s] = GREY
4: while Q ≠ ∅ do
5:    u ← EXTRACT-MIN(Q)
6:    colour[u] ← BLACK
7:    for each vertex v ∈ Adj[u] do
8:       RELAX(u, v, e, ρ)
9:       if colour[v] = WHITE then
10:         Q ← Q ∪ {v}
11:         colour[v] = GREY
12:      end if
13:   end for
14: end while
```

ALGORITHM 2: INITIALISE (G, s, t, ρ, K)

$\pi[v]$ denotes the predecessor node of v along the selected path, and NIL is the initial condition when it is empty before starting the path computation.

```
1: for each vertex v ∈ V do
2:    d[v] ← ∞
3:    dρ[v] ← ∞
4:    π[v] ← NIL
5:    colour[v] ← WHITE
6:    if v ∈ any of the K paths then
7:       colour[v] ← BLACK
8:    else
9:       Compute ρ[v] w.r.t. primary path
10:   end if
11: end for
12: d[s] ← 0
13: dρ[s] ← 0
14: colour[s] ← colour[t] ← WHITE
```

ALGORITHM 3: RELAX (u, v, e, ρ)

```
1: if dρ[v] > dρ[u] + ρ[v] then
2:    dρ[v] ← dρ[u] + ρ[v]
3:    d[v] ← d[u] + e(u, v)
4:    π[v] ← u
5: else if dρ[v] = dρ[u] + ρ[v] and d[v] > d[u] + e(u, v) then
6:    dρ[v] ← dρ[u] + ρ[v]
7:    d[v] ← d[u] + e(u, v)
8:    π[v] ← u
9: end if
```

— In the end, we obtain a primary path, and a set of backup paths in ascending order of their correlation weights w.r.t. the primary path.

The pseudo-code of the centralised NDM scheme is shown in Algorithm 1. It is similar to the Dijkstra's shortest path algorithm [Dijkstra 1959] with a modified RELAX procedure that appears in Algorithm 3. In Dijkstra where only the hop-count metric is used to select the routes, here both correlation ($d\rho$) and hop-count (d) metrics are jointly utilised. Based on (3), a node specific correlation factor $\rho(u) \in \{0, 1\}$ is assigned to each u that quantifies whether it is a neighbour to any node of the primary path. $d\rho[u]$ defines the cumulative ρ's from source s upto node u. Priority is first given to disjointedness (i.e., correlation factors) in choosing the nodes along a backup path. In case of equal disjointedness while considering two different nodes, the smaller hop-count route is preferred.

Next we prove that Algorithm 1 computes the least correlated backup path w.r.t. the primary path. The key is to show that each time a vertex u is coloured BLACK, we have $d\rho[u] = \delta\rho(s, u)$.

THEOREM 3.1. *Algorithm 1 runs on a weighted undirected graph $G = (V, E)$ with non-negative weight function ρ and source s, and terminates with $d\rho[u] = \delta\rho(s, u)$ for each $u \in V$.*

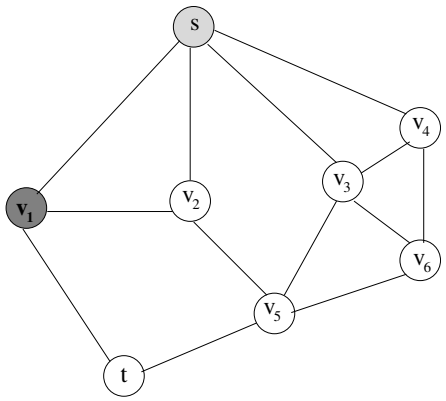
PROOF. See Appendix A.1. □

COROLLARY 3.2. *Algorithm 1 computes the least correlated neighbour disjoint backup path w.r.t. the primary path.*

PROOF. The corollary follows directly from Theorem 3.1. By replacing u by t we obtain, $d\rho[t] = \delta\rho(s, t)$ which verifies the computed backup path, $s \rightsquigarrow t$ as the least correlated. □

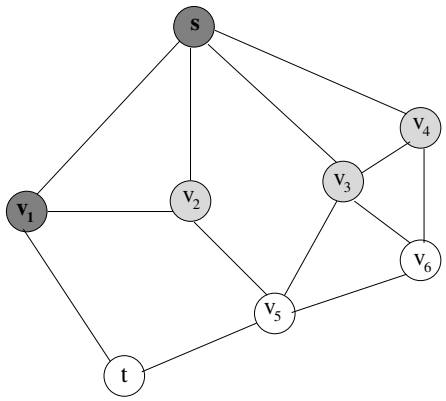
Corollary 3.2 is valid for any number of backup paths. For example, at first, algorithm 1 is called with $K = 1$ after the primary path is computed. Subsequently, the first backup path is obtained which certainly is the least correlated w.r.t. the primary path. For subsequent calls to the algorithm ($K > 1$), the additional backup paths computed will still be the least correlated ones given the sample space available. The reduced sample space in each call will only contain the vertices that are not part of the K paths computed previously.

3.2.1. Complexity Analysis. The following analysis provides an upper bound on the computational costs for running the complete NDM algorithm. It is similar to the complexity of Dijkstra's shortest path algorithm for a single source-sink pair. The complexity primarily depends on lines 4, 5, 7 and 8 of algorithm 1. If the EXTRACT_MIN procedure (i.e., priority queue) is managed with a binary heap, then the cost of retrieval of a minimum weight vertex (line 5) is $\mathcal{O}(\log_2 V)$. There will be $|V|$ such operations. All the edges E will be traversed in line 7, and for each traversal, the RELAX procedure's operation (implicit in algorithm 3) on the binary heap will cost $\mathcal{O}(\log_2 V)$. Therefore, for a single path calculation between source and sink, the upper bound can be computed as $\mathcal{O}((V+E)\log_2 V)$. If there exists K paths between the source and sink, algorithm 1 will be run for K times. Consequently, the running time is $\mathcal{O}(K(V+E)\log_2 V)$. This running time corresponds to a single node's identification of its primary and backup paths towards the sink using the global network topology information. Ford-Fulkerson's maximum flow variant was applied to compute the edge-disjoint multipaths [Ford and Fulkerson 2010]. The same algorithm can be applied for the node-disjoint multipath computation as well by using node-splitting technique [Cormen et al. 2001]. Their running time complexities are approximately $\mathcal{O}(KE)$.



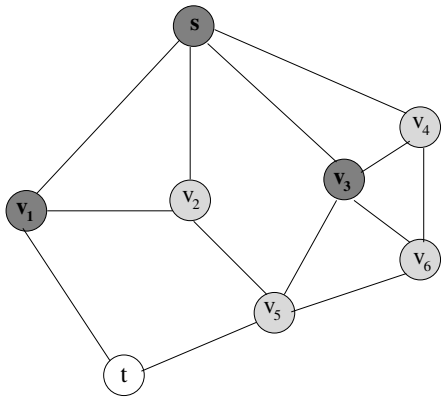
	s	v_1	v_2	v_3	v_4	v_5	v_6	t
d	0	---	MAX	MAX	MAX	MAX	MAX	MAX
$d\rho$	0.0	---	MAX	MAX	MAX	MAX	MAX	MAX
π	NIL	---	NIL	NIL	NIL	NIL	NIL	NIL

(a) After initialisation.



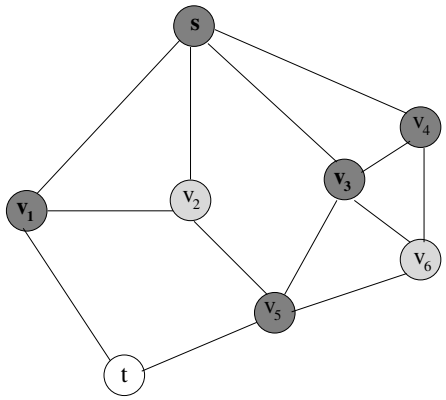
	s	v_1	v_2	v_3	v_4	v_5	v_6	t
d	0	---	1	1	1	MAX	MAX	MAX
$d\rho$	0.0	---	1.0	0.0	0.0	MAX	MAX	MAX
π	NIL	---	s	s	s	NIL	NIL	NIL

(b) After iteration 1: s's edges are relaxed.



	s	v_1	v_2	v_3	v_4	v_5	v_6	t
d	0	---	1	1	1	2	2	MAX
$d\rho$	0.0	---	1.0	0.0	0.0	0.0	0.0	MAX
π	NIL	---	s	s	s	v_3	v_3	NIL

(c) After iteration 2: v_3's edges are relaxed.



	s	v_1	v_2	v_3	v_4	v_5	v_6	t
d	0	---	1	1	1	2	2	MAX
$d\rho$	0.0	---	1.0	0.0	0.0	0.0	0.0	MAX
π	NIL	---	s	s	s	v_3	v_3	NIL

(d) After iteration 4: v_5's edges are relaxed.

Fig. 1: An illustrative example showing NDM algorithm's computation steps in order to find a backup path between source (s) and sink (t).

3.2.2. An Illustrative Example. Consider the network topology of Fig. 1 where the source and sink are denoted by s and t, respectively. We will explain the computation steps of a backup path using the NDM algorithm. Assume the primary path $< s, v_1, t >$ has already been computed. Fig. 1a depicts the network's state after the initialisation step

of algorithm 1. Note that, the dark shaded vertices are the ones that have already been explored while lightly shaded ones correspond to the nodes currently inside the priority queue, Q. The rest are the unexplored ones. Fig. 1b depicts the network's state after the first iteration of *while loop* at line 4 where the edges of the source (s) are traversed (or relaxed). The tables under each network state of the figures list some important metrics, e.g., the hop-count distance (d), correlation factor ($d\rho$), and the predecessor (π) associated with each vertex. During second iteration, the EXTRACT_MIN procedure of algorithm 1 might choose either v_3 or v_4 from Q. However, v_2 would not be chosen because of its high correlation factor ($d\rho = 1.0$). Suppose, v_3 is chosen, and the network's state after the iteration appears in Fig. 1c. During iteration 3 (the state after which is not shown in Fig. 1), vertex v_4 will be chosen. Note that, two other candidates v_5 and v_6 having the same correlation factor as v_4 can not be selected in this step. This is because v_4's hop-count distance metric ($d = 1$) is lower than theirs ($d = 2$). After iteration 3, either v_5 or v_6 could have been chosen. Suppose v_5 is chosen, and Fig. 1d represents the network state after iteration 4. Even if v_6 were chosen in this step, the obtained backup path would have been the same ultimately. In subsequent iterations, vertex v_6, v_2 and t will be chosen respectively that will yield no changes in the network states. The backup path $< s, v_3, v_5, t >$ can be retrieved by traversing the predecessors (π's) of the vertices starting from sink t.

Note that there exists another completely uncorrelated NDM path $< s, v_4, v_6, v_5, t >$. However, our algorithm selected the shorter distance path among the two. In other words, between two NDM paths having similar correlation factors, our algorithm would always choose the shorter distance path. If both the metrics (d and $d\rho$) are the same between two paths, then one is randomly chosen.

3.3. Evaluation Criteria

In this section, we first define the performance metrics that are used for evaluation purpose, and then briefly discuss the localised failure model adopted.

3.3.1. Performance Metrics

— *Packet Delivery Ratio (PDR):* It is defined as the ratio of the successfully received packets to the total number of packets sent. In the graphs of the results of Section 3.4 and 5, PDR is converted to percentages. This standard metric is a direct indication of the reliability of a certain protocol. The more packets are delivered successfully the more reliable the protocol is.

— *End-to-end latency:* It is the total delay that a successfully received packet experiences from the instant it is sent from the source until it is successfully received at the destination.

— *Overhead:* We categorise overhead as the average number of control messages exchanged in order to setup or maintain routes for communication. This parameter is even of more importance for failure-prone scenarios where frequent disruptions of the set paths may result in exchange of more control messages. Even though we kept energy consumption metric outside the scope of this paper, it is directly proportional to the quantity of the control traffic generated. Therefore, it also gives an indirect measure of energy efficiency of the protocol.

— *Average routing table size:* It is also an important parameter for memory constrained LLN devices. There are generally two different modes of operations according to RPL RFC [Winter et al. 2012] – storing and non-storing. Route entries are saved in the storing mode, and no storage is used for routing table in the other. We report this parameter for evaluation purpose since storing mode WSN nodes are used in our experiments.

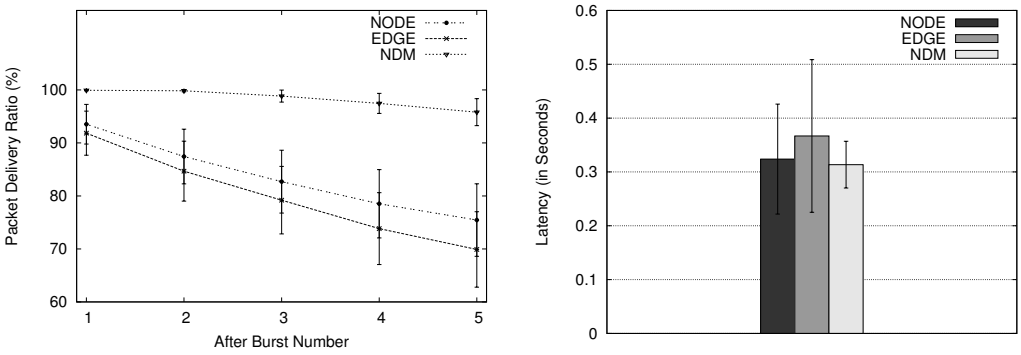
All the above metrics have been measured when node or link failures are introduced according to the model discussed in the next section. Node or link failures are more commonplace in low-power and lossy WSN than wired networks. NDM was conceptualised in order to provide resilience amidst such failures. Our aim is to measure these metrics in such scenarios in order to investigate NDM's practicality.

3.3.2. Failure Model. A localised failure model that was used in our previous work [Hossain et al. 2014] is adopted. This model corresponds to the failure of all nodes within a circular region of R_l. It attempts to model the idealised wave propagation of most physical phenomena [Ganesan et al. 2001] where the effect may only be observed within a specific region. Various types of activities such as interference or jamming inside a building affecting multiple nodes within an area, or other environmental effects such as rain fades or fire may destruct sensors confined to a specific region. In our model, the centre of the localised failure's region R_l is assumed to be uniformly distributed over the sensor field. Furthermore, we assume the number of such failures within a time interval is Poisson distributed with parameter λ_l. Both transient and permanent failures discussed in Section 1 can be modelled with this model. However, in our simulations, we assume once a sensor is affected by a failure event, they remain nonoperational during the remaining run-time. In addition, a variant of this model where the temporary node failures resulting only in intermittent connectivity loss within the circular region has been considered in testbed experiments of Section 5.1.2. The intermittent connectivity loss period is kept constant (30 sec) for each failure incident. *Isolated* failure model was considered in our previous work [Hossain et al. 2014] where each node may die independently of each other. We did not observe significant improvement of NDM over its node-disjoint and edge-disjoint multipath approaches in such failure scenarios. This was expected since NDM was conceptualised to fight against co-located node failures. However, we repeated the testbed experiment of Section 5.1.2 with this failure model as well.

3.4. Comparison with NODE and EDGE

To compare the various disjoint multipath schemes, a UDP application is run where packets are sent in bursts on a NS-3 platform [The NS-3 network simulator 2016]. This is to mimic response of the WSN to an event happening. The five bursts' timings are uniformly distributed over the simulation duration, and each burst consists of 128 packets. Localised failures are introduced with parameters, $R_l = 15$m and $\lambda_l = 3$. Unit disk model [Rappaport 2001] propagation characteristics inside an outdoor simulation area of $400m \times 400m$ is considered. Each node's transmission radius is fixed at $50m$ and 802.11b standard is used as the MAC protocol. 200 nodes are randomly deployed throughout the area where the source and sink are chosen randomly in each trial that are separated by $6 \sim 7$ hops. The optimised link state routing (OLSR) daemon is utilised to retrieve the neighbour information of each node. Afterwards the NDM algorithm is run inside a centralised entry (e.g., the source). The Ford-Fulkerson algorithm [Cormen et al. 2001] is utilised to centrally compute the node-disjoint and edge-disjoint paths.

Fig. 2 is constructed as the average of 100 simulation runs with 95% confidence interval. Only one backup path is used for all the NODE, EDGE and NDM schemes. The application sends packets through both the primary and the backup path at all times using source routing. In other words, two copies of the same packet is sent through two different paths. We have shown in [Hossain et al. 2014] that our NDM approach is more resilient than NODE or EDGE in presence of localised failures. The effect of multiple backup paths was also explored in [Hossain et al. 2014] where the resilience to failures increased monotonically with the number of backup paths for all three ap-



(a) Packet delivery ratio (PDR). (b) End-to-end latency.

Fig. 2: NS-3 Simulation Results

proaches. The results in Fig. 2a reaffirms this statement as NDM's packet delivery ratio (PDR) is the highest. Interestingly, latency-wise NDM also proved to be superior compared to NODE and EDGE as shown in Fig. 2b. NDM generated backup paths are generally longer than its NODE and EDGE counterparts [Hossain et al. 2014] where shorter backup paths (smaller hop-count metric) are selected. Therefore, NDM is expected to incur higher latency. However, our results show otherwise. The latency is defined as the time interval between sending both the packets from source to the earliest packet's arrival time at destination. We attribute the smaller end-to-end delay for NDM to the underlying MAC protocol's characteristics (e.g., 802.11b's carrier sensing mechanism) together with how the packet sending application operates. The application sends copies of the same packet through the primary and backup path at the same time. They might be competing for the same medium for NODE and EDGE whereas for NDM, the backup path is expected to be spatially separated, thereby might suffer smaller latency. Furthermore, the higher variance of EDGE can be explained using the same phenomenon as even shared nodes may appear along multiple paths resulting from EDGE protocol. EDGE might have produced paths sharing nodes (higher latency) in some runs, and in others, might have generated spatially disjoint paths (lower latency), which we believe has given rise to the resulting variation.

The other two performance metric parameters discussed in Section 3.3.1 are not considered here since source routing is used, and no failure detection and recovery mechanism is adopted. As a result, the average routing table size and the overhead metric (control messages generated to fix or maintenance of routes) are of no importance. However, all the four performance metric parameters are used for evaluation purpose when Contiki OS simulator and real testbed implementation are considered in Section 5.

4. DISTRIBUTED NDM

We proved the optimality of the centralised NDM algorithm in terms of least correlated neighbour disjoint backup path computation in the previous section. However, its operation depends on the assumption of knowledge of the whole topology by the source which may entail quite an expensive operation for resource constrained WSNs. Therefore, a distributed NDM scheme is conceived, and has been implemented on top of LOADng – a lightweight AODV reactive routing protocol. NDM is not constrained by the underlying routing protocol with which it may work. However, its route computation process needs to be modified to accommodate NDM for calculating the backup

paths. It could have been implemented on top of RPL as well. We chose LOADng because of its reactive nature that is characterised by incurring lower overhead, and simplicity. RPL is not selected since it already has some redundant path construction concept in its operation [Winter et al. 2012]. We compare both basic LOADng and RPL with our implementation which we term as "LOADng+NDM".

Distributed NDM utilises RREQ and RREP message exchanges of LOADng in order to come up with the primary and backup paths. During the route discovery phase, RREQ message is broadcasted from source to know the address of the destination. All the intermediate nodes insert their identifiers inside the RREQ message if it passes through them which ultimately reaches the destination. The destination delays the sending of RREP message until multiple RREQs are received that might arrive through different paths. Thereby, the destination can infer the primary path, i.e., by examining the hop count of different paths through which RREQs have arrived. The destination now creates two different RREP messages (unicast and broadcast), and sends them one after another. Algorithm 4 shows the processing of RREP messages inside the intermediate nodes. When a unicast RREP (RREP_U) is received, it is treated in exactly the same way as it would have been for a traditional AODV protocol. It is sent via the primary path where reverse routes were setup by the RREQ process. When a broadcast RREP message (RREP_B) is received, first the node checks if it is inside the primary path that is attached inside the RREP_B message by the destination. If not, it further checks if it is a neighbour to any node of the primary path, and update the correlation and hop-count metrics accordingly. It only rebroadcasts the RREP if the correlation metric yields an improvement, or the hop-count metric improves in case of equal correlation parameter. Otherwise the RREP_B is dropped. Similar to RREQ propagation, a sequence ID is also used to limit the broadcast of stale RREP_B messages. Eventually, both the primary and backup paths will be setup by the RREP messages.

Sometimes we have observed scarcity of backup paths created by our NDM algorithm in sparse networks. For example, if we apply algorithm 4 for Fig. 3's topology, no backup path will be obtained. Therefore, we relaxed the requirement that a backup path must not share any node with the primary path for our distributed NDM implementation. However, we still penalise such backup paths by increasing their correlation metric. In other words, rather than dropping RREP_B as in line 7 and 8 of algorithm 4, we replaced the two lines with $\rho[v] = 1$.

Measuring the resilience to failures in our experiments leads to reaction to route failures to be an important aspect to be handled by the routing protocol. Route-Error (RERR) message propagation of LOADng is implemented by taking the simplest approach among the multiple options specified in [Clausen et al. 2014]. An intermediate node sends RERR message towards the source when it is unable to send a packet (identified from link layer acknowledgements). If the source's route is not available inside the identifier node, a RREQ for the source node will be issued first, followed by the RERR towards it. The source, and all the intermediate nodes including the RERR identifier node then delete the route entry for the destination. For the next packet to be sent from the source, naturally, a fresh RREQ will be issued. On the contrary, LOADng+NDM will check for the backup path to be used first, and only issue a fresh RREQ if it is unavailable. For both approaches, it generally results in only the in-transit packet to be lost.

There is also a need for a periodic keep-alive control message exchange to retain the backup paths. The primary path will be active since packets will be passing through its nodes. On the contrary, the route entry for node 2 inside 4 of Fig. 3 will be purged after route holding time interval. Note that, node 4 is only inside the backup path 5-4-2-1 and no packet passes through it when the primary path is active. For this reason, a periodic keep-alive message needs to be sent along the backup path. For the

ALGORITHM 4: RREP processing at intermediate node, v after receiving the RREP from u

Initially all the $d\rho[v]$ and $d[v]$ at the intermediate nodes are set to maximum.

```
1: if RREP_U then
2:    Forward in usual way like AODV
3: else
4:    ρ[v] ← 0
5:    for ∀_x inside primary path except s, t do
6:       if x = v then
7:          Drop RREP_B {v inside the primary path, so drop RREP}
8:          return
9:       else if x is neighbour to v then
10:         ρ[v] ← 1 {if v is a neighbour to any node inside primary path}
11:      end if
12:   end for
13:   if dρ[v] > dρ[u] + ρ[v] then
14:      dρ[v] ← dρ[u] + ρ[v]
15:      d[v] ← d[u] + 1
16:      π[v] ← u
17:      Attach node v's identifier inside the path, dρ[v], d[v] and rebroadcast RREP_B
18:   else if dρ[v] = dρ[u] + ρ[v] and d[v] > d[u] + 1 then
19:      dρ[v] ← dρ[u] + ρ[v]
20:      d[v] ← d[u] + 1
21:      π[v] ← u
22:      Attach node v's identifier inside the path, dρ[v], d[v] and rebroadcast RREP_B
23:   else
24:      Drop RREP_B
25:   end if
26: end if
```

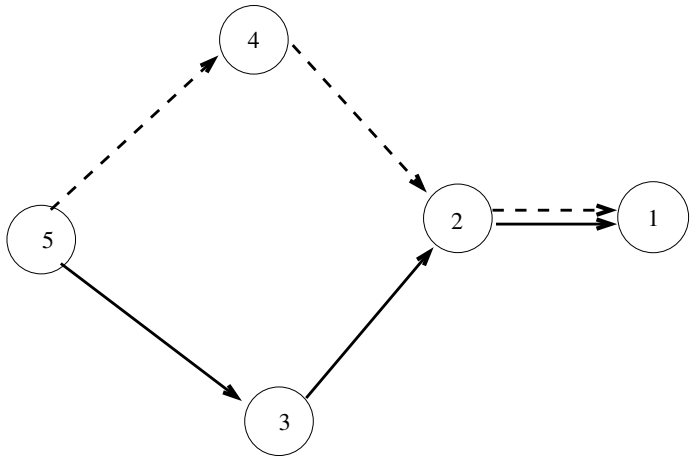


Fig. 3: Sample Topology. Primary path (5-3-2-1) is shown as solid lines whereas the backup path (5-4-2-1) is shown as dashed lines.

routing table entry, apart from the usual fields, the metric (e.g., hop count (HC)) and next-hop entries are duplicated for both the primary and backup NDMs whereas the NDM-specific correlation metric dρ is only added for the backup path.

5. EXPERIMENTS, RESULTS AND DISCUSSION

In this section, we outline the advantages arising from using NDM compared to the two well-known routing protocols for LLN, namely RPL and LOADng w.r.t. performance metrics defined in Section 3.3.1. The WSN applications discussed in the following are considered in presence of failure model that is explained in Section 3.3.2. We then explain the simulation and TWIST testbed environments, and the parameters that are varied to assess the various routing protocols' performances in the next section.

From WSN perspective, the application traffic pattern can be categorised as sensors to sink (multi-point to point (MP2P)), sink to sensors (point to multi-point (P2MP)), and a single sensor to another sensor or sink (point to point (P2P)) types. The research community is divided in their opinions about the superiority of RPL or LOADng since one is not seen to perform better than the other universally in all application scenarios [Yi et al. 2013; Herberg and Clausen 2011].

RPL operates with the assumption that MP2P network traffic is predominant whereas P2MP and P2P traffic is quite rare [Winter et al. 2012]. By incorporating such operating principle in the protocol design, RPL generally outsmarts other routing protocols in MP2P application scenarios [Vučinić et al. 2013] whereas LOADng is generally seen to perform better in P2P scenarios [Herberg and Clausen 2011]. Since we also compare NDM with these two popular WSN routing protocols, we did not confine ourselves in using only one particular application that may favour one or the other. As a result, the following three application scenarios are considered:

— *P2P:* In some application scenarios, a sensor to another single sensor or sink communication might be more common. For example, in the remote control application in building automation, more than 30% of traffic can be P2P [Martocci et al. 2010]. We consider one application scenario in our experiments where only communication between a single sensor and the sink takes place.
— *MP2P and P2MP:* Sensor applications utilising these types of traffic pattern were foretold to be the most common type by the RPL research community where even the P2MP traffic is assumed to be low compared to MP2P traffic [Winter et al. 2012]. In our experiments, we consider an application scenario where a few sensors periodically send packets to a single sink (MP2P), and the sink acknowledges the packet reception (P2MP).
— *Multiple P2P:* In the P2P scenario discussed above, we consider a application where only one P2P communication is active at a time. In this category, we consider the scenario when multiple P2P communications are active at the same time which might be quite common for distributed control, i.e., multiple pairs of sensor-sink communication would be prevalent inside the same network.

5.1. Simulation and Testbed Environments

Both simulation and real-world testbed experiments are performed for evaluating RPL, LOADng and our implementation LOADng+NDM considering the three different types of applications described above.

5.1.1. Cooja Simulation Environment. Contiki OS and its hardware emulator/simulator Cooja is used as a framework for LLN IPv6 routing simulation [Tsiftes et al. 2010]. We have compiled Contiki for the Tmote Sky platform and an IEEE 802.15.4 complaint radio, thereby taking into account the hardware constraints as well. The radio propagation model is Unit Disk Graph (UDG) where nodes are located in the Euclidean plane and are assumed to have identical (unit) transmission radii. The simulation parameters are summarised in Table I.

Table I: Cooja Simulation Parameters

Settings	Value
Testbed size	100×100m
Mote type (number)	Tmote Sky (30)
Transmission power	19 dBm
Wireless channel model	UDG with distance loss
Transport and network layer	UDP + μIPv6 + 6LoWPAN
MAC layer	non-slotted CSMA + ContikiMAC
Radio interface	CC2420 2.4 GHz (IEEE 802.15.4)
Application	Echo server and client
Packet arrival rate	once in 20 seconds (random)
Simulation time	10 runs – each 30 minutes long
Failure parameters	localised 15m radius poisson arrival with average 3

5.1.2. TWIST Testbed Environment. TWIST [Handziski et al. 2006] testbed spans over 3 floors of TU Berlin Academic Campus with an indoor office area of 1500 m^2 having both Tmote Sky and eyesIFX sensors (102 each). We have only used the 102 Tmote Sky sensors. The testbed consists of a central server (to give access to outside users), control PCs and gateway nodes that are all connected through a Ethernet back-channel. The gateways or so-called super nodes are Network Link Storage Units (NSLUs) running a customised Linux OS with USB interfaces. USB hubs and cables are used to connect to the actual sensor network for power-supply, programming and communication. To extend the reach of the network a combination of passive (up to 5m in length) and active (up to 15m in length) USB cables are used. Each USB cable endpoint is defined as a socket with a unique identifier in the management software giving the sensor's association and its geographical location. By sending a suitable USB message, it can control the power state of a given port in the USB hub, thereby turning on or off the power supply of any downstream device. These testbed attributes (i.e., turning the device on/off) enabled us to easily emulate the failure models discussed in Section 3.3.2. The testbed parameters are summarised in Table II.

Table II: TWIST Testbed Parameters

Settings	Value
Testbed size	$32 \times 15 \times 16$m
Mote type (number)	Tmote Sky (102)
Transmission power	19 dBm
Transport and network layer	UDP + μIPv6 + 6LoWPAN
MAC layer	non-slotted CSMA + ContikiMAC
Radio interface	CC2420 2.4 GHz (IEEE 802.15.4)
Application	Server and client (one way traffic)
Packet arrival rate	once in 20 seconds (random)
Run time	5 runs – each 10 minutes long
Failure parameters	localised 2m, 3m and 4m radius poisson arrival with average 3

5.2. Simulation Results and Discussions

The performance metrics of Section 3.3.1 considering the three types of applications mentioned previously for RPL, LOADng, and our implementation LOADng+NDM are calculated. The various metrics' results for P2P, MP2P & P2MP, and multiple P2P applications are shown in Fig. 4, 5 and 6, respectively. All the figures are constructed as the average of 10 simulation runs with 95% confidence interval (except Fig. 5b). For MP2P & P2MP, a sink and 5 other sensors are randomly chosen as sources in each simulation run. For multiple P2P applications, 5 P2P pairs are chosen randomly in each run; however, the geographical separation of each pair is kept the same. In the following, we provide the results and observations of RPL, LOADng and LOADng+NDM in terms of PDR, end-to-end latency, overhead, and average routing table size:

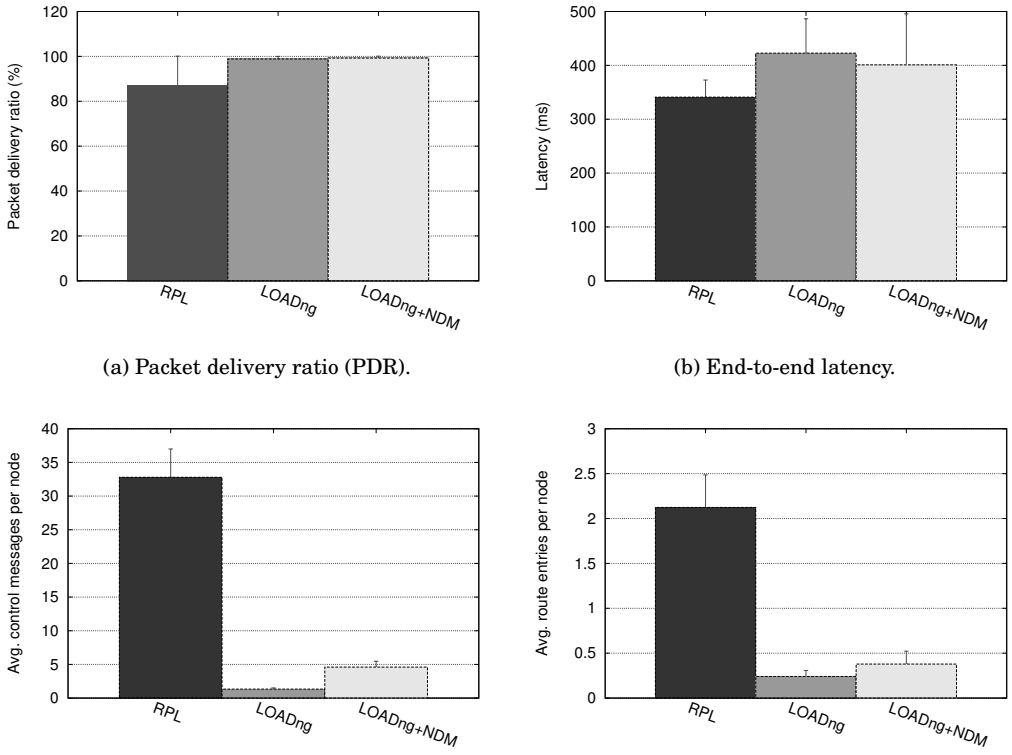
5.2.1. PDR. LOADng and LOADng+NDM perform better than the RPL for P2P applications (Fig. 4a), whereas RPL's packet delivery ratio is slightly higher than LOADng for MP2P & P2MP applications (Fig. 5a). For MP2P & P2MP, we considered both RFC 6550's suggested parameters that affect the control overhead generation, and also the default parameter setting that comes with the Contiki OS distribution [Contiki OS 2016]. They are listed in Table III. Failures are introduced according to our model where the sensors inside a particular radius are assumed to have failed. The number of such failure incidents obtained from a poisson distribution for a single simulation run is uniformly distributed over the total simulation time. We observed that RPL's response to failures is quite slow as its stale routes take time to be purged. On the contrary, in both LOADng and LOADng+NDM, the response is much quicker as RERR message propagates right after discovering the link/neighbour failures through missing link layer acknowledgements.

Table III: RPL Protocol Parameters

RPL Specification	DIO Min Interval	DIO Max Interval
RFC 6550	8 ms	2.3 hr
Contiki OS	4 s	1048 s

There are a few other points to be noted. Firstly, because of how NDM is constructed as described in Section 4, we could not apply it for MP2P & P2MP applications. NDM is constructed by finding the disjoint P2P backup path between a source and a destination w.r.t. their primary path. In case of multi-point application scenario, the routing tables at the intermediate nodes should be redefined for source-destination pair rather than for the destination only. Secondly, the root of the RPL DODAG and the sink is considered to be the same entity for a single P2P application. However, this assumption does not hold when multiple P2P applications exist inside the same network. Therefore, RPL is omitted for such application scenario even though it could be adopted in a complicated way or as an extension to RPL [Winter et al. 2012]. However, this property does not come with the standard Contiki OS distribution. Thirdly, for both P2P and multiple P2P applications, our LOADng+NDM is seen to perform slightly better than the standard LOADng (see Fig. 4a and 6a). We believe this is due to the availability of the backup path after failures that was constructed by NDM.

5.2.2. End-to-end Latency. It is computed by only considering the timestamps for successfully received packets at the destination. As can be seen from Fig. 4b and 5b, RPL offers the best performance as its routing table is computed beforehand whereas for both LOADng and LOADng+NDM, routes are computed on-demand that includes a route discovery phase delay. LOADng+NDM's performance is slightly better than



(a) Packet delivery ratio (PDR).

(b) End-to-end latency.

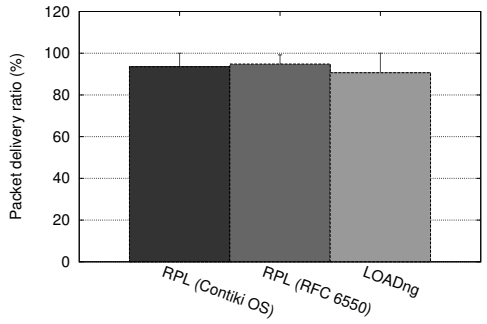
(c) Average control messages exchanged per node.

(d) Average routing table size.

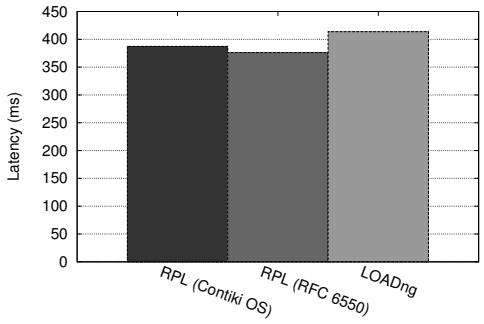
Fig. 4: P2P Application Results

LOADng for a single P2P application (Fig. 4b). This is due to the fact that for some failure scenarios, a backup path was available. As a result, no route discovery phase was instigated. On the contrary, LOADng's performance is slightly better than LOADng+NDM (Fig. 6b) for multiple P2P applications. We believe this is due to the existence of multiple traffic flows inside the network. This may cause an NDM backup path that is generally longer than a LOADng primary path to have shared links/nodes with some other separate P2P paths. This scenario does not generally occur for a single P2P flow. MP2P & P2MP scenario Fig. 5b comprises of various source-sink pair traffic separated by different geographic distances, therefore, no confidence interval can be shown. The different separating distances of the source-sink pairs are kept the same across the protocols for fair comparison though.

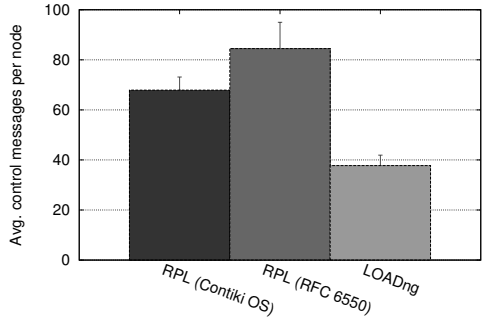
5.2.3. Overhead. Since RPL is a proactive protocol, it performs worse than the other two w.r.t. overhead for both P2P and multi-point scenarios. LOADng performs slightly better than our LOADng+NDM since a periodic keep-alive control message is introduced, and sent via the backup path. It ensures the backup path to remain operational when not being used. For example, consider the topology of Fig. 3 where packets are sent via the primary path. If no keep-alive message is used, after route holding time interval, the route entry for node 2 inside node 4 will be purged even though it is inside the backup path. However, sending periodic keep-alive message is only one possible solution. There could be other alternatives, e.g., if a node is inside a backup path, its backup routing entry is never invalidated. In that case, both LOADng and



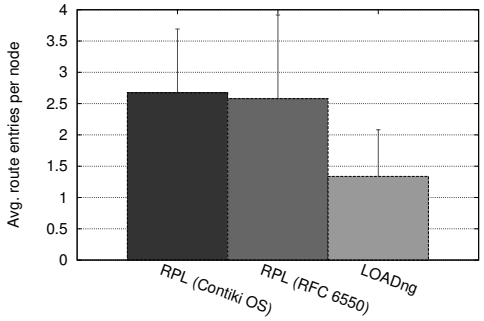
(a) Packet delivery ratio (PDR).



(b) End-to-end latency.



(c) Average control messages exchanged per node.



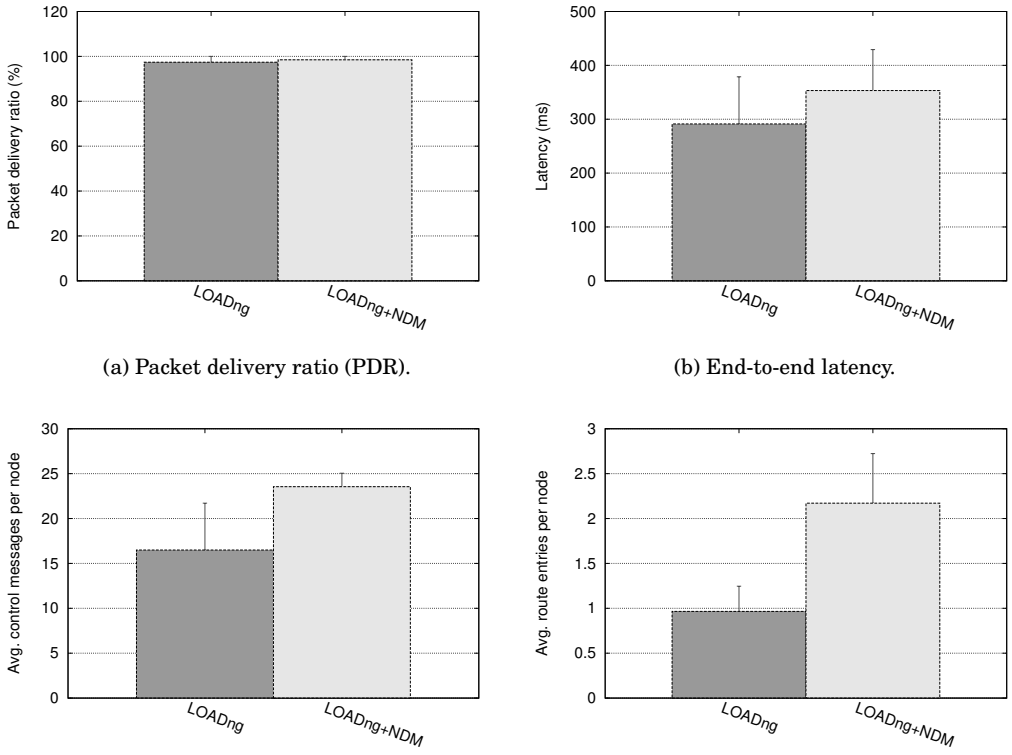
(d) Average routing table size.

Fig. 5: MP2P and P2MP Application Results

LOADng+NDM's overhead would almost be the same except for the additional $RREP_B$ control message that is only used during the route discovery phase.

For MP2P & P2MP, no significant difference was observed for both variants of RPL implementations in case of PDR and latency. However, the control overhead generated for RFC 6550 implementation is quite large compared to its Contiki OS counterpart. This can be explained from Table III's parameter settings as more control messages are expected to be emitted for RFC 6550 than Contiki OS because of the changing topology caused by failure-prone scenarios. For this reason, we only use Contiki OS's RPL implementation for the other P2P related experiments.

5.2.4. Average Routing Table Size. We select the storing mode of the LLN devices where the routing table is stored inside the node. Therefore, we believe the average routing table size may also affect such memory constrained devices' performance. At specific time intervals, the routing tables of all the nodes are dumped, and thereby the average number of routing entries per node are computed, and subsequently plotted as in Fig. 4d, 5d, 6d. We observe RPL's DODAG root to contain route entries for all the other nodes inside the network. The default maximum routing table entries per node is however restricted to 20 in Contiki OS implementation. This characteristics combined with proactive nature of RPL results in its larger average routing table size throughout the observation time compared to the other two where routes are entered on-demand basis (see Fig. 4d, 5d and 6d). Because of LOADng+NDM's additional routing entries for its backup routes, its average routing table entries per node is slightly higher than that of LOADng.



(a) Packet delivery ratio (PDR).

(b) End-to-end latency.

(c) Average control messages exchanged per node.

(d) Average routing table size.

Fig. 6: Multiple P2P Application Results

5.3. Testbed Results and Discussions

Fig. 7, 8 and 9 depict the PDR of a P2P application that is run in TWIST testbed environment [Handziski et al. 2006]. A source and a sink separated by the same distance ($\approx$ 3 hops) are randomly chosen in each run. The bars of the figures are the average of 5 independent runs with 95% confidence interval. The radius within which the nodes/links are made to fail both permanently and transiently using the localised model described in Section 3.3.2 is varied. We also present a set of results with independent failure model in Fig. 9. The number of node failures are kept the same as the other two models for each run to be consistent, but are made to fail independently.

There are quite a number of interesting findings that we touch upon in the following. First, the PDR generally tends to monotonically decrease for all the protocols as failure radius gets bigger. Second, RPL performs the worst for such a P2P application which also complies with the simulation findings discussed in Section 5.2.1. We observed that RPL could not even recover if the failure event affects its active path, i.e., no fresh route was discovered during the remaining run-time. RPL's performance gain observed in Fig. 8 compared to Fig. 7 is also a testament to this factor. This is because the active path of RPL is only affected transiently in Fig. 8 as it did not change. On the contrary, LOADng was seen to work around its broken path. LOADng+NDM gives the best performance because of its backup path availability when the primary path is affected by the both failure incidents as depicted in Fig. 7 and 8, respectively. Third, even though performance gain is observed for using NDM for independent failure model scenario too (Fig. 9), it is not as significant as the localised one. This complies with the

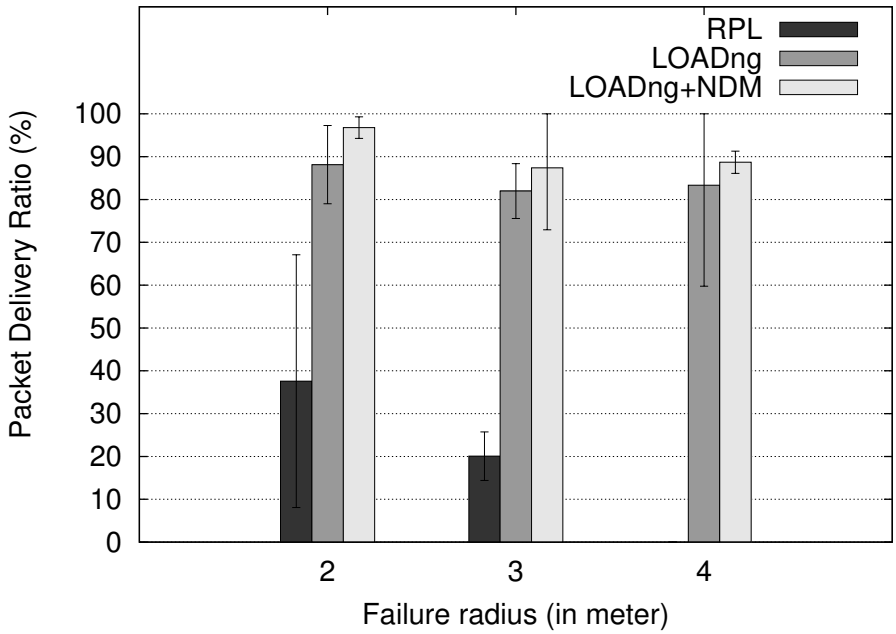


Fig. 7: Packet delivery ratio (PDR) in TWIST testbed experiment with varying failure radius, and permanent connectivity loss within the failure region.

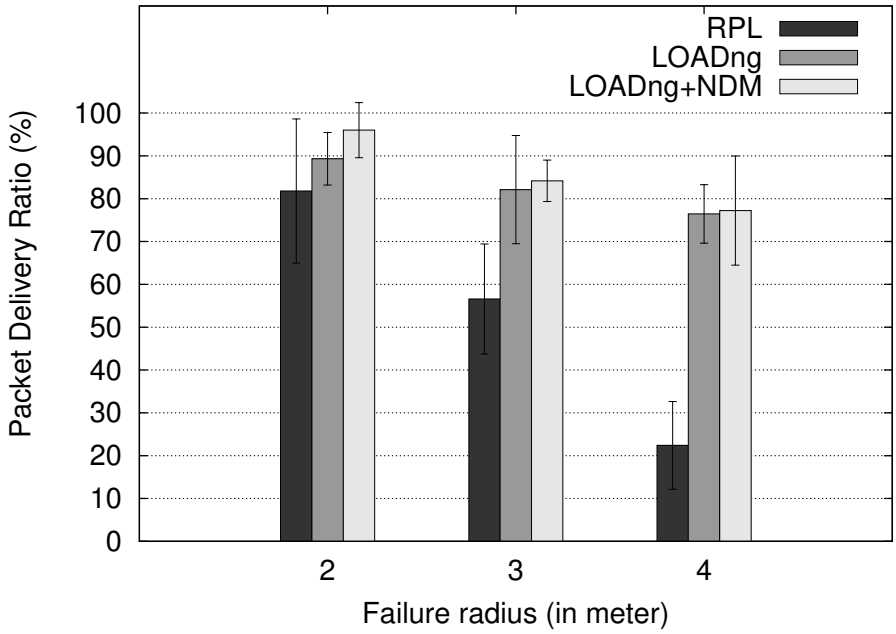


Fig. 8: Packet delivery ratio (PDR) in TWIST testbed experiment with varying failure radius, and intermittent connectivity loss within the failure region.

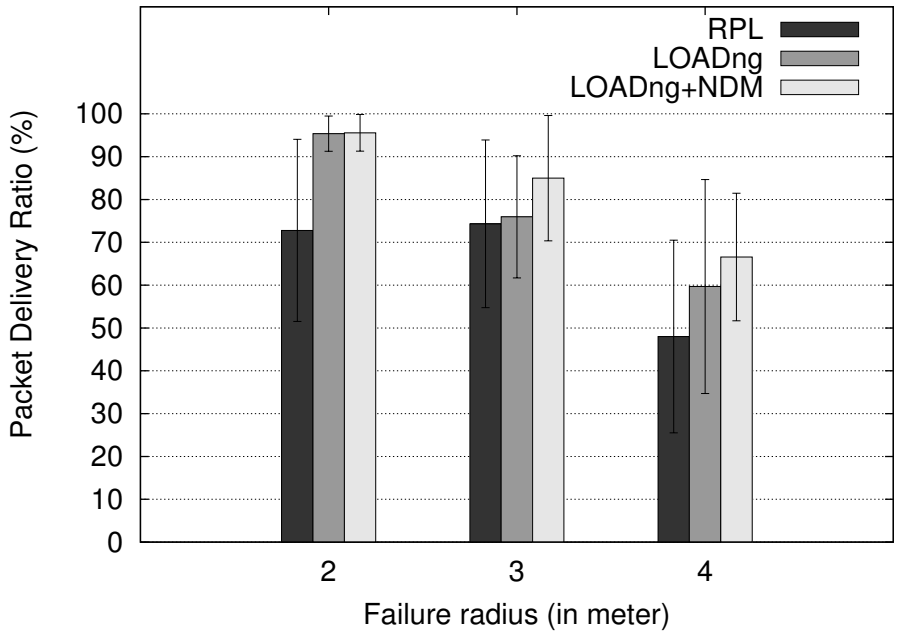


Fig. 9: Packet delivery ratio (PDR) in TWIST testbed experiment with independent failure model.

findings of our previous work [Hossain et al. 2014] where NDM was conceptualised to fight against co-located node failures more. Fourth, both overhead and routing table size were obtained from the source and sink which showed similar trend as our simulation observations. It was not possible to retrieve the information when all the nodes were activated in the logging procedure as the testbed used to get overwhelmed with the amount of data produced. Fifth, end-to-end latency could not be retrieved as the sensor clocks were not synchronised. This is a prerequisite condition for recording the sending time instant at the sender and receiving time instant at the destination. Finally, we could only conduct a few experiments because of the testbed's reservation policy of limited time per week for outside users, and also for the experimental time constraint.

6. CONCLUSION AND FUTURE WORK

A centralised Neighbour Disjoint Multipath (NDM) was first conceptualised in our initial published work [Hossain et al. 2014] utilising the spatial diversity among multiple paths to ensure robustness against localised poor channel quality or node failures. In this article, we prove its optimality and also show its superiority over NODE and EDGE in terms of throughput and latency amidst localised failure scenarios. We further describe a distributed NDM scheme adapting to the low-lower and lossy network characteristics, and its implementation on top of LOADng in detail. We compare its implementation, LOADng+NDM with two well-known routing protocols for LLN, namely RPL and basic LOADng w.r.t. some standard performance metrics. We consider three different application scenarios such as P2P, MP2P & P2MP, and Multiple P2P for the comparison purpose. Based on the results and observations, we conclude that for P2P applications, LOADng should be used instead of RPL. LOADng+NDM could further improve LOADng's performance in terms of latency and throughput. No significant

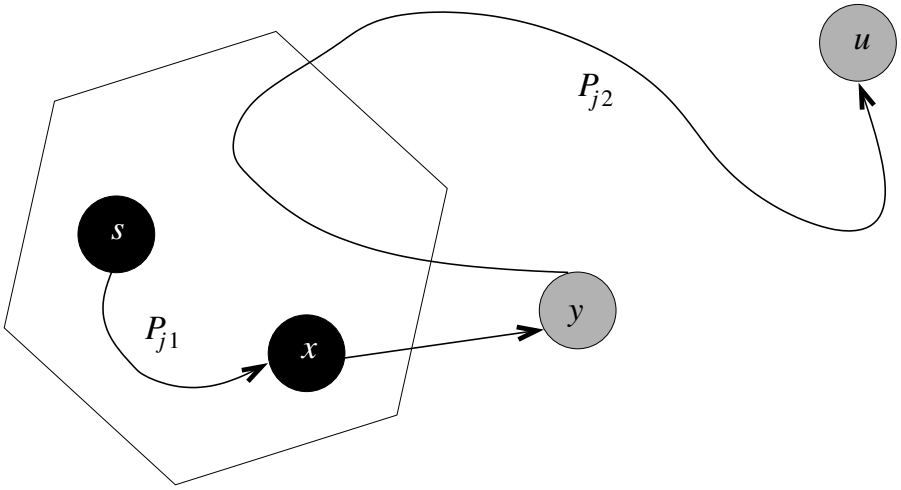


Fig. 10: NDM backup path P_j from s to u may be decomposed into $s \overset{P_{j1}}{\rightsquigarrow} x \rightarrow y \overset{P_{j2}}{\rightsquigarrow} u$. Vertices x and y are distinct, but we may have $s = x$ or $y = u$. Path P_{j2} may or may not reenter the set containing the vertices that are coloured BLACK.

difference between RPL and LOADng is noticed even in MP2P & P2MP applications in terms of throughput. From an overhead perspective, RPL does not perform well in any of the application scenarios whereas it performs superior from a latency perspective. It is also observed that RPL's performance can be improved by selecting configuration parameters carefully.

We foresee a few future work directions. NDM's backup path routing entry could be stored w.r.t. source-destination pair rather than the conventional destination oriented entry, thereby making it suitable for MP2P & P2MP applications. Subsequently, it may be implemented on top of RPL to inspect if it could improve RPL's performance. More insights are also required into the applicability of the appropriate routing protocol depending on the application need.

APPENDIX

A.1. Proof of Theorem 3.1.

During the start of iteration of the **while** loop at line 4 of Algorithm 1, $d\rho[v] = \delta\rho(s, v) = \infty$ for each $v \in V$ that are part of the primary path or the already constructed NDM backup paths. In other words, these vertices are not part of the current backup path computation. Now it suffices to show that for each vertex $u \in V \setminus \{V_1 \cup V_2 \cup \ldots \cup V_K\} \cup \{s, t\}$, we have $d\rho[u] = \delta\rho(s, u)$ at the time u is coloured BLACK. Here, K is the number of paths computed so far where V_1 comprises the vertex set of the primary path $(k = 1)$, and V_k comprises the vertex set of $(k-1)^{\text{th}}$ backup path.

We wish to show that in each iteration of the **while** loop, $d\rho[u] = \delta\rho(s, u)$ for the vertex u that is coloured BLACK. For the purpose of contradiction, let u be the first vertex for which $d\rho[u] \neq \delta\rho(s, u)$ when it is coloured BLACK. We must have $u \neq s$ since s is coloured BLACK during the first iteration, and $d[s] = \delta\rho(s, s) = 0$ at that time. Because $u \neq s$, there must be some path from s to u, for otherwise $d\rho[u] = \delta\rho(s, u) = \infty$ by the no-path property, which would violate our assumption that $d\rho[u] \neq \delta\rho(s, u)$. Therefore, there is at least one path from s to u.

Now consider Fig. 10 where both s and x are inside the set that contains the vertices that are already coloured BLACK, and y and u are inside the set that are still unex-

plored. An NDM backup path P_j from s to u can be decomposed as $s \overset{P_{j1}}{\rightsquigarrow} x \rightarrow y \overset{P_{j2}}{\rightsquigarrow} u$. It follows that $d\rho[x] = \delta\rho(s,x)$ when x was coloured BLACK since u is the *first* vertex for which $d\rho[u] \neq \delta\rho(s,u)$. Subsequently, by relaxing edge (x,y), we have $d\rho[y] = \delta\rho(s,y)$ (see Algorithm 3).

We try to obtain a contradiction to prove that $d\rho[u] = \delta\rho(s,u)$. Because y occurs before u on an NDM backup path from s to u and the weights are non-negative, we have $\delta\rho(s,y) \leq \delta\rho(s,u)$, and thus

$$\begin{aligned} d\rho[y] &= \delta\rho(s,y) \\ &\leq \delta\rho(s,u) \\ &\leq d\rho(u) \end{aligned} \tag{4}$$

However, because both u and y are unexplored when u is chosen, we have,

$$d\rho[u] \leq d\rho[y]. \tag{5}$$

Comparing (4) and (5), we have, $d\rho[y] = \delta\rho(s,y) = \delta\rho(s,u) = d\rho[u]$. Consequently, $d\rho[u] = \delta\rho(s,u)$ which contradicts our choice of u. We conclude that $d\rho[u] = \delta\rho(s,u)$ when u is coloured BLACK, and this equality is maintained at all times thereafter. □

ACKNOWLEDGMENTS

This research has been fully funded by the Irish Research Council and United Technologies Research Center Ireland Ltd. We acknowledge Szymon Fedor's contributions to the centralised NDM design and evaluation. We would like to thank Mališa Vučinić and Martin Heusse for contributing their "Link Reversal and Reactive Routing" protocol [La et al. 2013] code of Contiki OS which we modified to implement LOADng first, and then we implemented our protocol NDM on top of it. We would also like to thank the TWIST testbed [Handziski et al. 2006] team for giving us the opportunity to run experiments on their WSN testbed.

REFERENCES

I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. 2002. Wireless sensor networks: a survey. *Computer Networks* 38, 4 (March 2002), 393–422.

A. Annamalai and Vijay K. Bhargava. 1998. Analysis and Optimization of Adaptive Multicopy Transmission ARQ Protocols for Time-Varying Channels. *IEEE Transactions on Com.* 46, 10 (1998), 1356–1368.

A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr. 2004. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing* 1, 1 (2004), 11–33.

Thomas Clausen, Axel Verdiere, Jiazi Yi, Afshin Niktash, Yuichi Igarashi, Hiroki Satoh, Ulrich Herberg, Cedric Lavenu, Thierry Lys, and Justin Dean. 2014. *The Lightweight On-demand Ad-hoc Distance-vector Routing Protocol - Next Generation (LOADng)*. Internet-Draft draft-clausen-lln-loadng-12.

Contiki OS. 2016. The Open Source Operating System for the Internet of Things. http://www.contiki-os.org/. (2016). Accessed: 25/03/2016.

Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. 2001. *Introduction to Algorithms* (2nd ed.). McGraw-Hill Higher Education.

E. W. Dijkstra. 1959. A Note on Two Problems in Connexion with Graphs. *Numer. Math.* 1, 1 (December 1959), 269–271.

D. R. Ford and D. R. Fulkerson. 2010. *Flows in Networks*. Princeton University Press.

Deepak Ganesan, Ramesh Govindan, Scott Shenker, and Deborah Estrin. 2001. Highly-resilient, Energy-efficient Multipath Routing in Wireless Sensor Networks. *Mobile Computing and Communications Review* 5, 4 (2001), 11–25.

Vlado Handziski, Andreas Köpke, Andreas Willig, and Adam Wolisz. 2006. TWIST: A Scalable and Reconfigurable Testbed for Wireless Indoor Experiments with Sensor Networks. In *Proceedings of the 2nd International Workshop on Multi-hop Ad Hoc Networks: From Theory to Reality (REALMAN '06)*. New York, USA, 63–70.

Tian He, J. A. Stankovic, Chenyang Lu, and T. Abdelzaher. 2003. SPEED: a stateless protocol for real-time communication in sensor networks. In *Proceedings of the International Conference on Distributed Computing Systems*. 46–55.

W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. 2002. An Application-Specific Protocol Architecture for Wireless Microsensor Networks. *IEEE Transactions on Wireless Communications* 1, 4 (October 2002), 660–670.

Ulrich Herberg and Thomas Clausen. 2011. A Comparative Performance Study of the Routing Protocols LOAD and RPL with Bi-directional Traffic in Low-power and Lossy Networks (LLN). In *Proc. of the 8th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*. New York, USA, 73–80.

A.K.M.M. Hossain, C.J. Sreenan, and S. Fedor. 2014. A Neighbour Disjoint Multipath Scheme for Fault Tolerant Wireless Sensor Networks. In *Proc. of IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS), 6th International Workshop on Performance Control in Wireless Sensor Networks (PWSN)*. 308–315.

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

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

Chi-Anh La, Martin Heusse, and Andrzej Duda Grenoble. 2013. Link reversal and reactive routing in Low Power and Lossy Networks. In *Proc. of IEEE PIMRC*. 3386–3390.

S. Lindsey and C.S. Raghavendra. 2002. PEGASIS: Power-efficient Gathering in Sensor Information System. *Proceedings IEEE Aerospace Conference* 3 (March 2002), 1125–1130.

Moufida Maimour. 2008. Maximally Radio-disjoint Multipath Routing for Wireless Multimedia Sensor Networks. In *Proc. of the 4th ACM WMuNeP*. 26–31.

J. Martocci, P. D. Mil, N. Riou, and W. Vermeylen. 2010. Building Automation Routing Requirements in Low-Power and Lossy Networks. IETF RFC 5867. (2010).

G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. 2007. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. IETF RFC 4944. (September 2007).

B. Nath and D. Niculescu. 2003. Routing on a curve. *ACM SIGCOMM Computer Communication Review* 33, 1 (2003), 155–160.

M. Patil and R. C. Biradar. 2012. A survey on routing protocols in Wireless Sensor Networks. In *Proceedings of the 18th IEEE International Conference on Networks (ICON)*. 86–91.

Charles E. Perkins and Elizabeth M. Royer. 1999. Ad-hoc On-Demand Distance Vector Routing. In *Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*. 90–100.

Marjan Radi, Behnam Dezfouli, Kamalrulnizam Abu Bakar, and Malrey Lee. 2012. Multipath Routing in Wireless Sensor Networks: Survey and Research Challenges. *Sensors* 12, 1 (2012), 650–685.

T. S. Rappaport. 2001. *Wireless Communications: Principles and Practice, 2nd Edition*. Prentice Hall.

Siuli Roy, Somprakash Bandyopadhyay, Tetsuro Ueda, and Kazuo Hasuike. 2002. Multipath Routing in Ad Hoc Wireless Networks with Omni Directional and Directional Antenna: A Comparative Study. In *Distributed Computing*. Lecture Notes in Computer Science, Vol. 2571. 184–191.

Kewei Sha, Jegnesh Gehlot, and Robert Greve. 2013. Multipath Routing Techniques in Wireless Sensor Networks: A Survey. *Wireless Personal Communications* 70, 2 (2013), 807–829.

Lanny Sitanayah, Kenneth N. Brown, and Cormac J. Sreenan. 2014. A fault-tolerant relay placement algorithm for ensuring k vertex-disjoint shortest paths in wireless sensor networks. *Ad Hoc Networks* 23 (2014), 145–162.

K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie. 2000. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications* 7, 5 (October 2000), 16–27.

Jenn-Yue Teo, Yajun Ha, and Chen-Khong Tham. 2008. Interference-Minimized Multipath Routing with Congestion Control in Wireless Sensor Network for High-Rate Streaming. *IEEE TMC* 7, 9 (Sept 2008), 1124–1137.

The NS-3 network simulator. 2016. A discrete-event network simulator. http://www.nsnam.org. (2016). Accessed: 25/03/2016.

J.W. Tsai and T. Moors. 2007. Interference-aware Multipath Selection for Reliable Routing in Wireless Mesh Networks. In *Proc. of IEEE MASS*. 1–6.

Nicolas Tsiftes, Joakim Eriksson, Niclas Finne, Fredrik Österlind, Joel Höglund, and Adam Dunkels. 2010. A Framework for Low-power IPv6 Routing Simulation, Experimentation, and Evaluation. *SIGCOMM Computer Communication Review* 40, 4 (2010), 479–480.

M. Vučinić, B. Tourancheau, and A. Duda. 2013. Performance comparison of the RPL and LOADng routing protocols in a Home Automation scenario. In *Proc. of IEEE WCNC*. 1974–1979.

A. Willig. 2005. Redundancy concepts to increase transmission reliability in wireless industrial LANs. *IEEE Transactions on Industrial Informatics* 1, 3 (2005), 173–182.

T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander. 2012. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. IETF RFC 6550. (March 2012).

Kui Wu and Janelle Harms. 2001. Performance Study of a Multipath Routing Method for Wireless Mobile Ad Hoc Networks. In *Proc. of MASCOTS*. 99–107.

Jiazi Yi, T. Clausen, and Y. Igarashi. 2013. Evaluation of routing protocol for low power and Lossy Networks: LOADng and RPL. In *Proc. of IEEE Conference on Wireless Sensor (ICWISE)*. 19–24.

Y. Yu, R. Govindan, and D. Estrin. 2001. *Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks*. Technical Report UCLA/CSD-TR-01-0023. UCLA Computer Science Department.