

C3P-RPL: A collaborative and proactive approach for optimal peer to peer path selection and sustenance.

Anusha T (✉ aa5293@srmist.edu.in)

SRM Institute of Science and Technology

Pushpalatha M

SRM Institute of Science and Technology

Research Article

Keywords: RPL, N2N / D2D / M2M, P2P, Shortest paths, Node betweenness centrality

Posted Date: April 4th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1505695/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

C3P-RPL: A collaborative and proactive approach for optimal peer to peer path selection and sustenance.

Anusha T and Dr Pushpalatha M

Department of Computing Technologies, SRM Institute of Science and Technology,
Kattangulathur, Chennai, India.

*Corresponding author(s). E-mail(s): aa5293@srmist.edu.in;
Contributing authors: pushpalm@srmist.edu.in;

Abstract

Internet of Things is evolving from information gathering platforms into collaborative systems where smart devices actively interact with each other in a seamless manner. For instance, Internet of Robotic Things is envisioned to provide augmented solutions through collaboration of varied smart devices and robots. These visions revolve around the ability of smart devices to directly communicate and cooperate with each other in real time. In this context, this paper is an attempt to study RPL's point to point routing that creates multi-hop paths between peer nodes. This standard routing protocol is known for robust and failsafe upward paths but its peer to peer (P2P) routes are reported to be sub optimal. This work assesses P2P performance of RPL's storing mode in a network of new generation devices having higher memory. Further, a Collaborative and Proactive Peer to Peer (C3P) path selection and sustenance approach is proposed where root node collates incremental topology from collaborative nodes and disseminates optimal single source shortest path trees $SPT(n)$. A progressive node betweenness centrality score ensures spread out paths. Minor topology changes are accommodated through incremental node and edge updates to targeted $SPT(n)$ locally. Storing SPTs in intermediate nodes reduces storage and packet size. Through simulations and testbed experiments, it is proven that C3P-RPL improves simultaneous peer to peer communication between all the nodes. Specifically, the path length is reduced by 30% and subsequently the network latency drops by 65% in an experimental testbed of 47 nodes, making it suitable for collaborations.

Keywords: RPL, N2N / D2D / M2M, P2P, Shortest paths, Node betweenness centrality

1 Introduction

Internet of Things has become omnipresent as we are embracing it openly for the various sophistication that it offers. Right from our homes to public spaces, personal assistance to industrial automation, the type of devices employed and their applications are expanding rapidly. New age micro-controllers are taking IoT to varied

fields with their augmented capabilities like bigger memory, higher computational power, faster caches, improved security and their ability for customizable specializations [1]. They facilitate newer solutions in myriad areas like Smart City, automation and e-commerce in Industrial IoT, collaboration in Internet of Robotic Things (IoRT), failsafe communication in Internet of Medical Things (IoMT) etc [2–4]. Point to point communication in

a Internet of Things network plays a vital role in bringing efficient collaborations between devices.

RPL is the standard routing protocol that empowers large number of constraint nodes to form a connected network [5] that finds many applications in home automation [6, 7], industrial automation [8] and urban environmental monitoring [9]. It employs ICMPv6 [10] control messages to form a multi-hop mesh network and a trickle timer [11] to control the distribution of these control messages. A central root node initiates the formation of Destination Oriented Directed Acyclic Graph (DODAG) with an DODAG Information Object (DIO) message. The DIOs trickle down with a rank metric and facilitate other nodes to form shorter acyclic paths towards the root node. Though the protocol was designed to cater multi-point to point (MP2P), point to multi-point (P2MP) and point to point (P2P) types of communication, it is known that the P2P routing paths are suboptimal. Non-storing mode RPL, always routes peer to peer packets through the root node and the storing mode RPL routes them through common ancestors (with the worst case through root node). Past research papers have flagged the memory requirement of RPL's storing mode [12, 13] operation and pointed out that it is not scalable. Hyung-Sin Kim et. al, in their survey [14], points out the availability of new generation micro-controller units (MCU) in recent times and prompts research community to move beyond the boundaries of constrained nodes.

To alleviate this P2P problem, Internet Engineering Task Force (IETF) extended RPL's specification to form point to point routing paths on demand, in a reactive manner [15]. A sender node wishing to find P2P routes, initiates a transitory directed acyclic graph (DAG) discovery by declaring itself as the root node for the DAG. Like the alignment to original root node, all nodes orient themselves towards the starter node. By reversing the routes obtained, the starter node gets best P2P routes to all other nodes in the network. To form a complete P2P mesh, all nodes in the network have to initiate this P2P route discovery. This method incurs huge control overhead when all the nodes want to communicate to all the peer nodes. In contrast to the reactive P2P path discovery, a few recent works propose an alternate centralized approach where root node

initiates neighbour graph collation [16, 17]. Armed with the neighbour graph and their link status, they calculate shortest paths between nodes. A requester node gets the shortest route from the root node, caches the same and sends the P2P packets by attaching source routing headers (SRH). SRH contains the entire route to traverse. This approach is not optimal for seamless P2P communication as each node has to wait for a P2P route before initiating the actual packet delivery. Then the nodes have to store the entire path for each route in cache. The increased packet size of this SRH approach requires more energy in transmitting it and / or limits the payload size of a P2P packet. Though the usage of SRH seems unavoidable for pure non-storing node, the same does not hold good for storing mode of or hybrid mode. Even in the non-storing mode, the storage requirement of multiple routes could be optimized.

This paper furthers the centralized scheme of peer to peer route discovery and proposes a simple method to reduce storage space, packet size and delay. A collaborative and proactive P2P RPL (C3P-RPL) is proposed with centrality assisted path selection algorithm and a distributed approach for route storage and sustenance. The significant characteristics of C3P-RPL are,

- **Collaborative:** Nodes initiate P2P route discovery by sharing local topology assessment with the root node.
- **Proactive:** Single source shortest path trees ($SPT(n)$, $n \in V$, rooted at n) computed at the root node are disseminated back. It has optimal routes to all nodes and are installed at each node in a proactive manner.
- **Accurate topology:** The topology is assessed asynchronously using received signal strength indicator (RSSI) metric at both ends of an edge.
- **Optimal path selection:** Best $SPT(n)$ is selected among the available single source shortest path trees (SSSPT) using a progressive node betweenness centrality metric for effective traffic distribution.
- **Low memory:** Storing the edges of $SPT(n)$ requires less memory $2*(|V|-1)$ when compared to storing entire routes ($x*(|V|-1)$, where x being the path length of routes).
- **Less overhead:** Table based routing at intermediate nodes facilitates smaller packet size

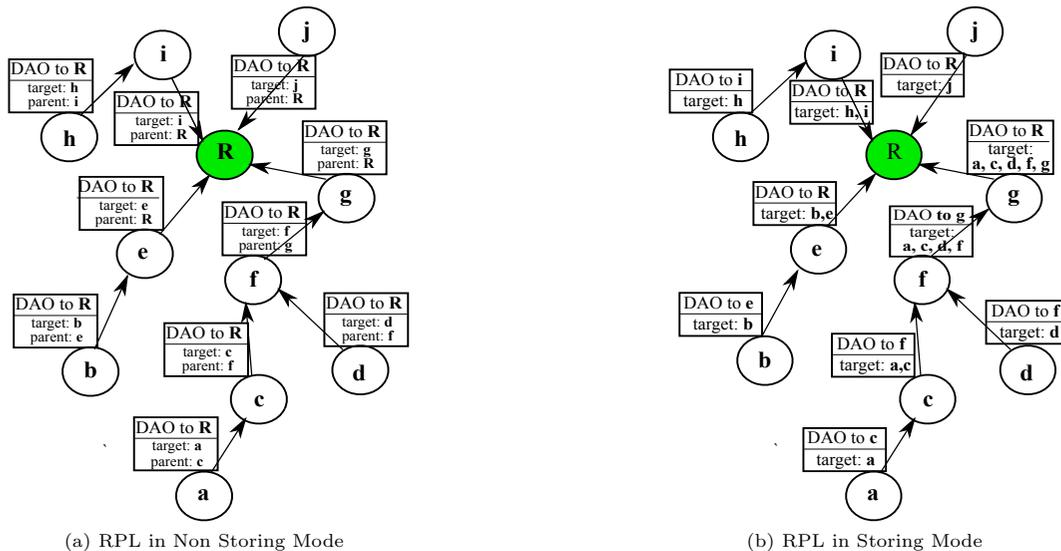


Fig. 1: RPL forming downward routes using Destination Advertisement Object control message

when compared to source routing which has entire route in it header.

2 RPL's Path Selection

RPL supports peer to peer traffic flows in both the storing and non-storing modes of operation. Downward routes are the basic requirement for identifying paths to peer nodes as P2P packets move upward until a route is available and then moves downward towards the destination. This section elaborates the nuances of the RPL's peer to peer route discovery.

2.1 Non-storing mode operation

RPL initiates route formation by propagating a DODAG Information Object (DIO) control message from the root node. The DIO has a rank metric that indicates the closeness of a node to the root. Each node calculates its rank metric by adding a path metric to its parent's rank and advertises the same in its DIO. In a RPL network, rank is smallest for the root node and increases as the distance from root node increases. A node selects a neighbour node as its preferred parent whose rank is the smallest among all the neighbours. So the upward route is formed by simply following the preferred parent at each node. This

forward route discovery is common for both the mode of operation.

RPL utilizes the Destination Advertisement Object (DAO) control message for constructing downward routes. Fig. 1a illustrates route formation process in RPL's non-storing mode. When a node selects a parent as preferred parent, it sends a DAO in the upward direction. The difference lies in the addressing of the DAO and its processing. Let node 'a' choose 'c' as parent. The DAO from a has its global address in the target option field and 'c' as parent address in the transit option field. It can also add multiple parents with their preference level. This DAO is unicasted to root node and forwarded through the preferred parent. None of the parents along the route would process the DAO but simply forwards it upward until it reaches the root. 'c' sends an independent DAO with its address as prefix and 'f' as the parent's address. Upon receiving the DAO's from the nodes, the root constructs the downward routes through the parent. When a node switches its parent, a DAO with the new parent's address is sent to the root node. So, all P2P data packets are forwarded in upward direction using the preferred parent as next hop towards the root node. The root finds the downward route and stores the same in its routing table. When a P2P packet arrives, it appends a Source Routing Header(SRH) containing the entire route and forwards the packet to

the next hop in downward direction. Every intermediate node in downward route processes SRH header, retrieves next hop, removes itself from the header and forwards the packet further down until it reaches the destination node.

2.2 Storing mode operation

The storing mode uses the same DAO control message with slight variation for forming downward routes. After parent selection, a node unicasts the DAO to its preferred parent and intermediate nodes store the routes. The figure in Fig. 1b illustrates P2P route discovery in a RPL network. For instance, node 'a' selects node 'c' as its preferred parent. It adds its global address as prefix in the target option of the DAO message and then unicasts the DAO to the preferred parent 'c'. Node 'c' on receiving the DAO, knows that a downward route exists to the sender node 'a' and saves the route 'a' to its route table. Unless 'c' is not a root node, on receiving a DAO from a child node, it should send an upward DAO with multiple consolidated prefixes or the single received prefix. Hence 'c' is not root, it unicasts a DAO to its parent *f* with addresses of 'a' and 'c' in the target option. Subsequently, 'f' knows that a downward route exists for 'c' and also a route exists for 'a' through 'c'. When the number of DAO's generated in these two modes are quantified against each other, it may look like the storing mode contributes more towards the control overhead. But it is not true from the transmission stand point since storing mode transmits DAO to a parent that doesn't travel any further and a new consolidated DAO taking its place. But DAOs in the non-storing mode needs to be forwarded until it reaches the root and the same DAO is transmitted at every hop.

Thus, a node receiving a DAO in storing mode, gets the destination address of a route from the target option and the next hop address from the sender address of the DAO packet. All intermediate nodes record the routes and utilize the same for forwarding data packets. When a parent switch happens, the node switching the parent sends out an upward no-path DAO to the previously selected parent. It is further forwarded in the upward direction to clear the old route. After the no-path DAO, a normal DAO is scheduled to

the newly selected parent. During packet delivery, a node operating in storing mode consults its route table. If an entry is present, the next hop address is fetched and the packet is forwarded to that next hop. If a route is not available for a destination, the node sends the data packet upward (through preferred parent) until it encounters an ancestor node with a route. In worst cases, the packet would go upto the root node as it is the common ancestor for all the nodes.

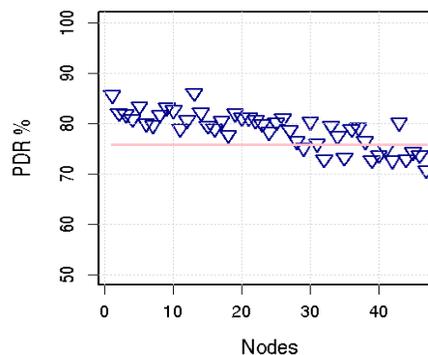


Fig. 2: RPL's storing mode performance in a testbed of 47 nodes having enough memory for routes.

In general, RPL's peer to peer routes are shorter in the storing mode than in the non-storing mode, as all packets need not go through the root node. However they are not optimal as they are forwarded along the DAG. When all nodes have sufficient memory to store the routes, RPL's delivery reliability is found to be 78.64% in storing mode of operation. Graph in Fig. 2 shows the packet delivery ratio for a RPL network in a testbed having 47 nodes where all nodes sends P2P data packets to all other nodes. The delivery ratio reduces as packets travel along the DAG creating congestion along few branch heads. Though all the nodes have sufficient memory to store routes, packet drops indicate that it is necessary to find short and distributed P2P routes.

3 Relevant work in literature

Downward routing forms the building block for point to point routing and many works concentrated on improving the scalability for downward routing even with the limited memory available in

the constrained nodes. According to survey in [14], only 11% of the research papers in RPL focuses on downward route formation and much less on the P2P routes. Multiple survey papers on RPL [18, 19] report that the P2P communication suffer from non-availability of memory in storing mode and long sub-optimal routes in non-storing mode of operation. M. Zhao et. al., in their survey [20] highlight that the communication delay and congestion are obstacles for successful node to node communication. There are past research works like multi-path routing [21, 22] where packets are forwarded through multiple parents using a path life cycle index / braided paths. Reliable path routing [23] uses multiple factors of parent nodes and QU-RPL [24] forwards packets based on the queue availability of parents. Though they show progress in reducing congestion in the upward direction, they might end up with increased parent switches. Frequent parent changes are known to increase control overhead and reduce reliability. Also they forward packets along the DAG and hence their peer to peer paths would still be longer as against this proposed work where it selects shortest yet traffic free paths.

As significance of peer to peer communication became prominent, the standardization of RPL is extended to include a reactive P2P route discovery with P2P-RPL in RFC 6997 [15]. Each node initiates optimal route discovery by declaring itself as a transient root node. E. Baccelli et. al and M. Zhao et. al, in [25, 26], studied the performance of P2P-RPL in simulations and testbed experiments. The testbed study confirmed that the P2P-RPL forms shorter point to point paths between nodes than the routes formed by RPL's non-storing mode. There is no documentation on the control overhead involved or energy consumption of P2P-RPL. It is only reported that the percentage of routes that pass through root node reduces significantly from 75% to 15%. But it is clear that this approach is intense in control overhead and is not suitable when all nodes wish to communicate with all other peers.

In order to improve downward routing DualMOP-RPL [27], combined the usage of both storing and non-storing modes in a single instance to improve scalability. Their proposed scheme made it mandatory to send parent's address in the transit option of DAO to make it inter-operable. However packets still have to flow along the DAG

and in turn encounter crowding along few nodes. OSR [28], is an opportunistic source routing protocol that was suggested as an alternate to RPL's non-storing mode. It identifies alternate downward paths opportunistically and compresses the source routing header with the use of bloom filters for lengthier routes. However bloom filters are known for false positives and may not be accurate. An energy conserving region-based ER-RPL [29] where a subset of nodes participate in route discovery as against the traditional method of holistic participation. The simulation in network simulator NS3 [30] shows improved performance when compared to P2P-RPL and RPL's non-storing mode for 100 nodes. However, the P2P traffic flow considered is linear with a route request response model and is not suitable for simultaneous P2P packet transmissions.

There are few works which go down the broadcast and multicast methods for improving the downward routing. A DIO overhearing based mechanism proposed by ERPL [31] which improves both the storing and non-storing mode by adding a route to the direct neighbours. Though this reduces the length of P2P routes for a subset of nodes, other nodes follow the original longer paths. There were research proposals that improved P2P routing by reducing the memory requirement of RPL's storing mode operation. MERPL [32], proposes to decouple a node and the table storage by using the child's resources to store more routes. In simulation, it showed improvement in performance for a network having nodes with low memory. D-RPL [33] proposed a multi-cast channel subscription in case of non-availability of routes. Whenever routes are available, it resorts back to the original RPL behavior. It records that the maximum limit of 60 entries in sky notes is not sufficient when the network is scaled up.

In contrast to the improvements made in RPL, there were alternate methods of route discovery for low power lossy networks (LLN). IoTorii [34] proposes a hierarchical addressing scheme to arrive at a local address for nodes in such a way that reflects the topology of the network. The nodes can gather multiple local addresses depending on the position of the node and uses the same for routing a packet to any destination. The evaluation is performed on cooja motes and shows reduction in table entries and control overhead.

However, there are no inference on cases of node failures that might impact the performance of the scheme.

Recently, there are couple of research works that improve P2P routes by proposing a centralized scheme. NG-RPL [16] and HRPL [17] proposes to collect the neighbor graph from all the nodes of the network, constructs shortest paths between nodes and uses source routing for data delivery along the shortest P2P routes. The energy conservation reported is significant in non-storing mode as they are currently following a longer path to root and then down to destination. Here too, the simulations do not consider a simultaneous flow of traffic between all nodes.

These problems were reported from the research works carried out on severely constrained nodes having low memory and computational capabilities. As far as the knowledge of the authors, there are no studies on the peer to peer packet delivery performance of storing mode in a network where all nodes have the capacity to store the routes. This paper is an attempt to do that and finds that even with sufficient memory, RPL suffers lower reliability, longer paths and higher latency. In cases of simultaneous P2P traffic between all nodes, it does not provide satisfactory results due to factors like congestion induced in the branch heads and collision along the long paths.

4 Optimal Peer to Peer Path Selection in C3P-RPL

The proposed approach offers couple of advantages over the current centralized approaches. The space required to store the set of P2P routes is made minimal by storing a simple single source shortest path tree (SSSPT) at each collaborative node. Subsequently, by storing $SPT(n)$ at each intermediate nodes, the need for the SRH header is eliminated. This allows for smaller packet sizes and saves energy when compared to SRH approach.

While forming SSSPTs, a node betweenness centrality score is applied to select a shortest yet distributed peer to peer routing path. SSSPT on node n is denoted as $SPT(n)$ and is rooted at n , is stored at node n and has $|V| - 1$ edges. Whereas storing individual routes in cache would

require space for $x * |V| - 1$ nodes, where x is the average path length of the routes from n . When there are minor changes to the network, the process of collation and dissemination is not repeated. Instead, individual targeted updates are carried out on the the stored SSSPTs in an incremental manner to accommodate node and link inclusions to the network.

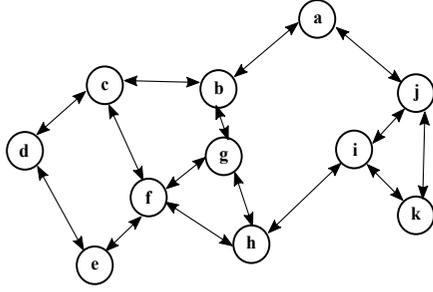
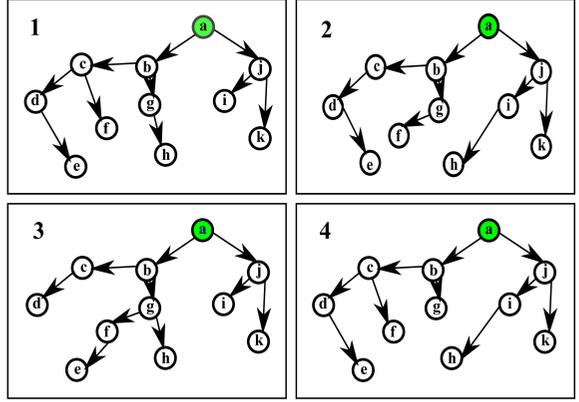
The proposed scheme has three distinct parts, namely,

- Collation of topology and construction of best possible single source shortest path trees (at root node).
- Extraction of routes from a shortest path tree (at each individual nodes).
- Reconstruction of a shortest path tree (at targeted nodes) after new node and link additions.

4.1 Collation of network topology and construction of SSSPT

A single source shortest path tree is a subgraph having smallest number of edges connecting the source node to all other nodes. It is formed such that the edges in $SPT(n)$ form shortest paths from the vertex n to all other vertices in the graph $G = \{V, E\}$. The graph G is assembled at the root node by collating incremental topology information from the collaborating nodes. It is suitably merged with the root node's routing table. Every collaborative node figures its positioning in the network by inspecting the received signal strength indicator (RSSI) metric for the packets received from neighboring nodes. A neighbour with a good RSSI value is considered as peer node and every collaborative node shares its peer information with the root node. An edge is considered bidirectional if nodes present on both the ends identify each other as peers, otherwise it is considered unidirectional. The root node merges its routing table and the collated topology information to arrive at graph G .

There may be multiple SSSPTs available for a single source vertex. A progressive node betweenness centrality metric is employed to select an shortest path tree that is spread out whose edges lie in minimal number of shortest routes. This ensures that the P2P data traffic is distributed across different nodes to the maximum possible extent and reduces congestion. Fig. 3a shows a

(a) Graph G 

(b) Shortest path trees

Fig. 3: Single source shortest path trees. (a) A directed graph G having 11 vertices and 15 edges (b) Possible single source shortest path trees rooted at vertex a .

graph $G = (V, E)$ with $|V| = 11$ and $|E| = 15$. For instance, multiple shortest path trees can be formed for the source node a as shown in Fig. 3b. Though all of them are shortest path trees having same path length, the SPT in 2 of Fig. 3b distributes P2P traffic effectively.

Node betweenness centrality (NBC) of a node is a measure of the number of shortest paths that passes through it. In a graph having a set of vertices V , let $\sigma(a, z)$ represent the total number of shortest paths between any two vertices a and z . Let $\sigma(a, x, z)$ denote the number of the shortest paths between a and z that pass through the vertex x . Then the node betweenness centrality for the vertex x is given by,

$$NBC(x) = \sum_{a \in V, z \in V; a, z \neq x} \frac{\sigma(a, x, z)}{\sigma(a, z)} \quad (1)$$

The summation happens over all the distinct pairs of vertices in the graph. In C3P-RPL, P2P traffic flows only on one chosen path between any two nodes. So essentially $\sigma(a, z)$ is 1 and $\sigma(a, x, z)$ is either 1 or 0, depending on whether the node x lies in the shortest path between a and z . So the equation 1 becomes

$$C3P_NBC(x) = \sum_{a \in V, z \in V; a, z \neq x} \begin{cases} 1 & \text{when } \sigma(a, x, z) \text{ is } 1 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

This progressive NBC score of a node is updated while forming single source shortest path trees with the number of shortest paths in which it features. When multiple shortest paths with different predecessors are available, the node having a small NBC score is chosen as the predecessor. This ensures that an edge or link is reused minimally. This is important for RPL network having a radio duty cycling (RDC) layer which allows a node's radio to sleep for most of the time to conserve energy. Multiple nodes forwarding data traffic through a single node is a major reason for packet loss in RPL network. Generally, shortest paths are identified with a breadth first search algorithm (BFS) [35]. However, this proposed work creates an optimal shortest path tree for each node using the NBC scores. The algorithm presented in 1 combines BFS and NBC scores for selecting an distributed uet shortest route between any two nodes.

This scheme offers shorter paths when compared to RPL's storing mode and saves energy by reducing the number of radio hops required for a packet to reach its peer node.

Implementation specifics

The design of C3P-RPL touches upon multiple aspects of routing to improve P2P routing. Foremost of all, the link local address in the format of fe80::xxxx is chosen for the nodes where the last 16 bit of the address is derived from the MAC address of the node. This short addressing mechanism

Algorithm 1 Centrality assisted P2P path selection algorithm for an optimal shortest path tree

Require: List having nodes and nbrs

```

for each src in List do
  src.level = 1
  src.pred = 0
  cur_level = 1
  for n in List with n.level = cur_level do
    Gather nbrs from n.nbrlist
    for nbr in n.nbrlist do
      if nbr.level is 0 then
        /*Edge (n,nbr) not in SPT*/
        nbr.level = cur_level+1
        nbr.pred = n
        Increment nbc for n
        Add edge (n, nbr) to SPT
      else if nbr.level is cur_level then
        /*Edge is already in SPT*/
        /*Another shortest path*/
        /*Choose pred with small nbc*/
        if nbr.pred.nbc > n.nbc + 1 then
          Del (nbr.pred, nbr) from SPT
          Decrement nbc of nbr.pred
          Add edge (n, nbr) to SPT
          Increment nbc of n
        end if
      end if
    end for
  end for
  Increment cur_level by 1
end for
Disseminate SPT to src
end for

```

is adopted for the ease of information collation and dissemination. The topology of the network is formed from the list of peers obtained from the collaborative nodes. The topology information object (TIO) is an Internet Control Message Protocol (ICMP6) packet that carries the topology information from the collaborative nodes to the root node. The format of a TIO is shown in Fig. 4 and has the number of peers listed followed by their short addresses.

Construction of $SPT(n)$ for each n occurs in the root node and are essentially minimum spanning trees having smallest number of edges connecting a source node with all other nodes. Once a $SPT(n)$ is created, it needs to be disseminated to the node n . A P2P Route Information Object (PRIO) ICMP6 message is utilized to send

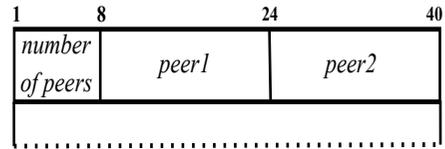


Fig. 4: Topology Information Object (TIO) as base object in an ICMP6 control message

the SPT back to each node n . As an SPT would have edge pairs for all the nodes in the network, it is split into multiple messages to comply the payload size of a IEEE 802.15.4 packet. The figure in Fig. 5 shows the format of the base object of the ICMP6 control message having a sequence number, number of edges in the SPT followed by the edge pairs.



Fig. 5: P2P Route Information Object (PRIO) as base object in an ICMP6 control message

The SPT is stored in the nodes as a list of edges for deriving next hops and are utilized for forwarding a P2P packet to its destination. The control overhead increases slightly but when compared to the overall DIS, DIO and DAO overhead, this is negligible.

4.2 Extraction of routes from a single source shortest path tree

A single source shortest path tree is the most favorable method to store the routes as it occupies less space and the same can be locally updated for the changes in the network. A route can be extracted from a single source shortest path tree by backtracking. It is a repetitive process that starts with a destination node in the SSSPT and traverses back until the path reaches the source node. The next hop node is the node having the source node as its predecessor node. The edge list has ordered (parent, child) pairs that represent the edges of SSSPT in the increasing order of path length starting with the source node. For instance, the ordered list of edges for the SSSPT shown in 2

of Fig. 3b would be (a, b), (a, j), (b, c), (b, g), (j, i), (j, k), (c, d), (g, f), (i, h) and (d, e). Here node *b* and *j* has the predecessor as the source node *a*, meaning that they are direct neighbours to *a*. There is a shortest route to *c* through *b*, denoting that the next hop for *c* is neighbor *b*.

Algorithm 2 Extract routes from a single source shortest path tree

Require: Ordered Edgelist, Source node *S*
for *n* in Edgelist **do**
 route_dest = *n.dest*
 next_hop = *GET_HOP*(*S*, *n.pred*, *n.dest*)
 (*route_dest*, *next_hop*) pairs are stored in table.
end for
function *GET_HOP*(*S*, *cur_pred*, *cur_dest*)
 if *cur_pred* is *S* **then**
 return *cur_dest*
 end if
 GET_HOP(*S*, *cur_pred.pred*, *cur_dest*)
end function

The algorithm specified in 2 extracts next hops for forwarding a P2P packet through the shortest route. With the availability of the extracted next hops, a peer to peer route is available before the start of P2P communication. This is advantageous for simultaneous peer to peer data packet transmission as there is no route request / reply concept.

4.3 Reconstruction of SSSPT for new node and edge inclusions

Shortest paths between nodes change as new nodes are added to the network. When the new node has links to more than one existing node, new paths might form between existing nodes through the new node. So reconstruction of an SPT becomes essential during new node inclusion or link formation. Performing collation and dissemination process for every edge or node inclusion is not feasible as it involves exchanging multiple TIOs and PRIOs. Hence a viable alternate is to update an existing SPT that is stored in an individual node for node and edge inclusions. This approach is equivalent to the incremental shortest path solution where existing shortest paths

are evaluated for impact and updated. G. Ramalingam et. al and A. Slobbe et. al, in their works [36, 37] provide bounded ways to incrementally update shortest paths.

Let *G'* (Fig. 6a) be the graph after adding a new vertex *z* and six new edges (*a,z*), (*z,a*), (*i,z*), (*z,i*), (*h,z*), (*z,h*) to the Graph *G* in Fig. 3a. Instead of considering *G'* as a new graph and constructing SPTs from the start, an alternate method is to incrementally update the existing SPTs. In each SPT, the affected shortest paths are identified and the edges are updated accordingly. The diagram in Fig. 6b illustrates the updates to SPT that is rooted in vertex *a*. Incremental up-gradation of SPT is a two fold process where the first step is to add the new node to the existing shortest path tree at the appropriate position. The next step is to identify the existence of a path that is shorter than the existing ones. When there is a new shortest path available through the new node, the predecessor of the impacted destination node is changed to the new node. The algorithm 3 has two separate functions for node insertion and edge updates. After the incremental update, the SPT has the shortest paths from the source node to all other nodes. When there are multiple node additions, the process is repeated for each node in a sequential manner.

4.3.1 Implementation specifics

Incremental updates to an existing SPT is limited to the nodes that are very near to the newly added node. The rationale is that the C3P-RPL nodes have next hops stored at each intermediate node and updating nodes closest to the new path is advantageous than updating nodes that are far off. In addition to sending a TIO to the root node, a node can be configured to broadcast it to the immediate neighbours. When a SPT is already installed in the neighbouring node, it automatically follows through the incremental update process. The TIO control message provides the incoming and outgoing edges of a new node. The algorithm in 3 adds the new node to the shortest path tree and updates existing shortest paths to nodes that are on the outgoing edges of the new node. Upon receiving a new TIO in root, it updates the node list, generates a new SPT rooted on the new node and sends PRIO control messages back to the new node. In essence, root

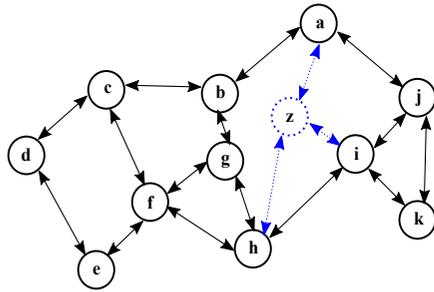
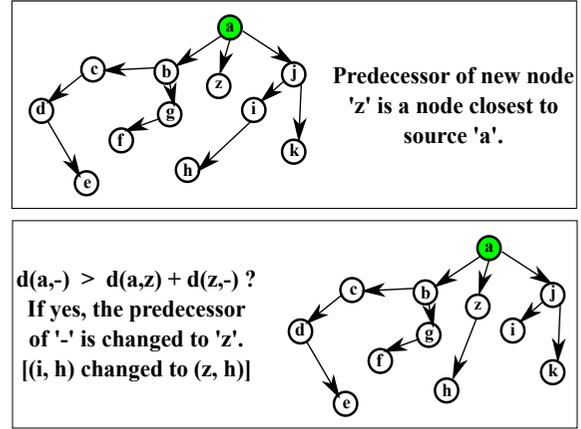
(a) Graph G' (b) Incremental update at node a

Fig. 6: Incremental Shortest Path. (a) The graph G' with a new node 'z' and six new edges included to G (b) The SPT at node a after the incremental update

Algorithm 3 Incremental update to a single source shortest path tree

Require: Ordered Edgelist, Source node S , New node N and outgoing edges of N

```

function ADD_NODE( $N$ ,  $N.nbrlist$ )
   $short\_dist = MAX\_VALUE$ 
  for  $nbr$  in  $N.nbrlist$  do
    if  $dist(S, nbr) < short\_dist$  then
       $short\_dist = dist(S, nbr)$ 
       $best\_pred = nbr$ 
    end if
  end for
  Add ( $best\_pred$ ,  $N$ ) pair to Edgelist
end function

```

```

function UPDATE_PATH( $N$ ,  $N.nbrlist$ )
  for  $nbr$  in  $N.nbrlist$  do
     $old\_dist = dist(S, nbr)$ 
     $new\_dist = dist(S, N) + 1$ 
    if  $new\_dist < old\_dist$  then
      Del ( $nbr.pred$ ,  $nbr$ ) from Edgelist
      Add ( $N$ ,  $nbr$ ) to Edgelist
    end if
  end for
end function

```

node delivers an optimal SPT to the new node and the existing SPTs in the neighbouring nodes undergo incremental updates locally. As it is not feasible to update all the existing SPTs frequently for topology changes, this targeted approach is proposed. This selective update approach is found

to give a good P2P packet delivery in case of small network changes. Complete reconstruction of entire set of shortest path trees is preferred for major network revamp or periodically and can be handled through configuration settings.

5 Experimental Evaluation

Both simulation and testbed experiments are performed to verify and quantify the benefits of the proposed strategy. The simulation study is undertaken using cooja simulator and the real-time testbed evaluation is performed in FIT IoT-LAB's [41] Lille site.

5.1 Network specification

The RPL network is customized to support 50 nodes. All nodes are routers, capable of forwarding data traffic and are made collaborative to find out the maximum possible benefit. The traffic is point to point and all nodes in the network transmit peer to peer data packets simultaneously to every other node in the network. The implementation of C3P-RPL pre-allocates memory in the order of $network_size * Max_neighbours$ in the root node and $network_size$ in all other nodes for storing the topology and the shortest path tree respectively. $Max_neighbours$ is taken as 20 for this study. The results are labeled as RPL and C3P against the respective protocols. To understand the behavior

of the proposed approach under new node additions, the experiments are repeated in the same network with $N - 3$ nodes starting at time t and 3 randomly selected nodes starting at $t + 12$ minutes (N being the number of nodes in the network). The corresponding results are labeled as RPL' and C3P' respectively. The following metrics are collected for evaluation,

- PDR: The packet delivery ratio as a percentage of peer to peer data packets received to that of the packets sent.
- Latency: The average time taken for a P2P packet to reach the destination node from a sender node.
- Hops: The average peer to peer path length between any two peer nodes in the network.
- Control Overhead: The average number control messages transmitted by a node. This includes DIS, DIO, DAO, TIO and PRIO.
- Duty Cycle: The percentage of time for which CPU and Radio was active against the total run time of the experiment.
- Traffic distribution: The number of P2P data packets that passes through each node.

5.2 Simulation study in cooja

The network chosen for testing is a grid topology having 50 wismote nodes. Wismote nodes have a TI MSP430 series 5 [38] processor and a TI CC2520 [39] 2.4GHz IEEE802.15.4 transceiver. ContikiRPL in Contiki operating system [40] is chosen as the RPL implementation and C3P-RPL changes are made on top of it. 49 nodes are placed in a 7x7 grid and the root node is placed above this grid at the top middle. The distance between nodes in the grid is 30m and the transmission range and interference range is the default 50m and 100m respectively. So corner nodes have two neighbors, nodes along the rim have 3 neighbours and all others have four neighbours. Each node in the grid transmits a data packet every 30 second and the packets are unicasted to all other nodes in the network. All nodes send and receive peer to peer data packets simultaneously and the simulation is run for one hour.

5.2.1 Result discussion

The graph in Fig. 7a exhibits the packet delivery ratio of all the nodes in the network. The root

node do not send any data packets and facilitate only the topology collation and route dissemination. The average pdr is at 88.8% for RPL and improves to 95.8% for the proposed C3P-RPL. Even though all the nodes in RPL have enough memory to hold routes, the nodes experiences packet loss. The probability of packet loss increases with path length as with every link there is a chance of collision and congestion. The maximum number of MAC layer retransmissions is set at 5 for low energy operations and packets are dropped after five failed attempts. In addition, RPL has to send packets upward until it reaches a common ancestor. So the nodes at the junction points experience congestion. C3P-RPL reduces packet loss by minimizing the path length of each route by routing along shortest paths and also reduces loss due to congestion by distributing traffic. The node betweenness centrality score coupled with shortest paths works well in minimizing loss due to congestion. The overall interquartile range is small for C3P-RPL which shows that all nodes perform equally well in contrast to RPL. In a network where node or edge additions happen after the DAG is formed, RPL's reliability is found to be little lower at 86.8% than the normal network. This is corresponding to the overhead increase seen in 7d by the new nodes to get attached to the DAG.

The graph in Fig. 7b displays the average path length or the number of hops between any two peer nodes in the network. The average path length between nodes in the grid is 6.8 for RPL and is reduced to 4.6 when shortest paths are employed for P2P communication in C3P-RPL. There is a reduction of around 31.5% in the path length between peers. Some of the nodes in RPL has a path length greater than ten as the packets travel along the DAG from one branch to another branch of the grid. When new nodes are introduced in network after it is stabilized (12 minutes from the start time), results in a slow progress of convergence in RPL. The path length of C3P' after new node inclusion is slightly higher as the shortest route updates are targeted and limited to the neighbouring nodes. But this strategy keeps the control overhead lower and maintains the reliability whereas RPL' has a low reliability in the new network.

Figure Fig. 7c shows the average time taken for a P2P packet to travel between two peers in

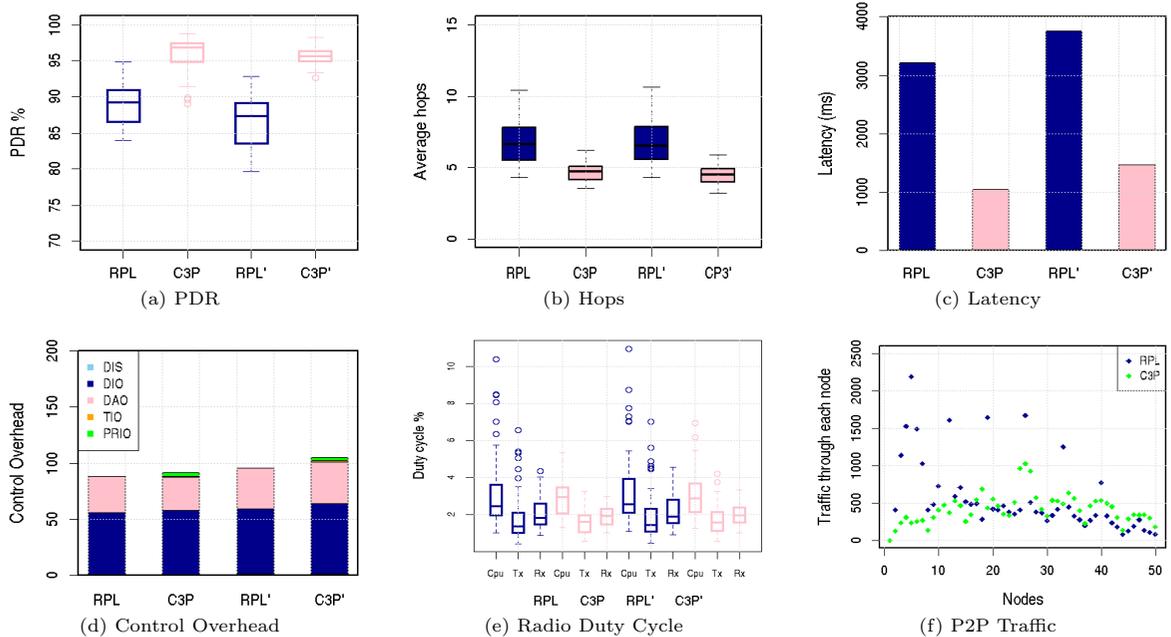


Fig. 7: Simulation results for P2P traffic in a 7x7 grid in cooja simulator (a) Packet delivery ratio (b) Path length of the P2P routes (c) Packet Latency (d) Number of control packets transmitted (e) Percentage Radio Duty Cycle (f) Number of data packets passing through each node.

the network. The latency is very high for RPL because the nodes farthest to the root node have to cross many intermediate nodes before reaching destination. With the reduction in path length and congestion, C3P-RPL reduces the travel time from 3219.6 ms to 1048.6 ms. This is a 67.4% of drop in network latency. The graph shows a 28.3% increase in latency for C3P' when compared to C3P because the incremental shortest path updates are performed only in neighbouring nodes and not universally across all the nodes. There is a trade-off involved in increasing the control overhead or increasing the path length of few P2P routes. However when compared to RPL', there is a considerable drop of 61.1% in latency for C3P' which justifies this targeted update strategy.

The averaged control overhead of a node is quantified in the Fig. 7d and as expected, it is slightly elevated for the proposed C3P-RPL. The expected transmission count (ETX) metric is recalculated after every successful data packet transmission and a parent switch happens whenever the path metric improves beyond a threshold. C3P-RPL install new routes through all the neighbours and as a result shows a small increase in

DIO generation. However, it can be seen that the newly added TIO and PRIO control messages are very minimum and does not add much to the control overhead. The overhead for C3P' is higher than C3P as three newly added nodes transmit more control packets to become part of the existing DAG. The preferred parent mechanism can be fine tuned for out-of-DAG packet forwards but is considered out of scope for this work.

The chart displayed in Fig. 7e gives the percentage of time for which the cpu, radio transmitter and receiver are active against the total running time of the simulation. A drop in the duty cycle is experienced for the proposed strategy as it reduces the number of hops required for the peer to peer routes. Though the computation is higher for the proposed approach, it reduces the number of intermediate nodes which has to process a incoming packet. It is imperative to note that the power consumption of a radio is much higher than the cpu. There are many outliers shown for RPL which indicates that few nodes that are junction heads need to forward more packets which results in energy depletion. It can be inferred

Table 1: RPL’s configuration parameters

Network	47 nodes
Run Time	3600 seconds
MAC technique	CSMA
Radio duty cycling	ContikiMAC
Channel check rate	8 HZ
Transmission power	-17 dbm
Receiver sensitivity	-72 dbm
Maximum route table entries	50

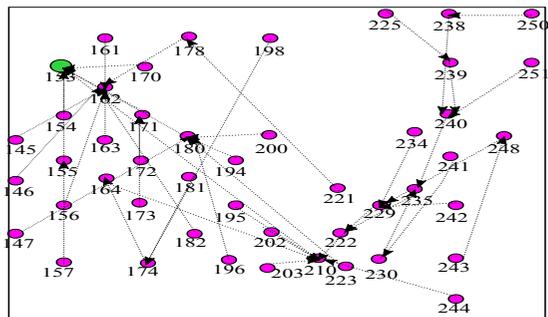
that C3P-RPL is more suitable for optimal energy utilization than RPL.

The number of data packets that is transmitted through each node directly depicts the P2P traffic distribution and is shown in the figure Fig. 7f. As it is a scatter plot, data points belonging to RPL and C3P-RPL alone are pitched against each other. The blue dots represents RPL and shows an uneven traffic distribution. Few branch heads cater to heavy traffic while others take in less traffic. In contrast, the green dots represent C3P-RPL where the traffic is evenly distributed. It is clear that C3P-RPL’s data distribution using shortest paths and NBC score provides optimal peer to peer shortest paths between nodes.

5.3 Experimentation in FIT IoT-LAB testbed

FIT IoT-LAB provides multiple testbed sites with hundreds of nodes for testing LLNs and multi-hop networks. They have atleast 20 different set of boards with different MCUs, radios and supports multiple operating systems. For this study, nodes having ARM Cortex M3 MCU with Atmel radio [42] are chosen. These nodes have 64kb of RAM and 256kb of ROM [43]. The positioning of the chosen 47 nodes in Lille test bed is as shown in Fig. 8.

In order to create a multi-hop network, the transmission power of the M3 nodes are set to a low -17db so that nodes do not directly connect to the root node. Table 1 lists the various configuration parameters that are set for RPL. The *UIP_CONF_MAX_ROUTES* parameter is set to network in order to allocate sufficient space for route tables. The channel check rate is set to a low 8Hz so that the radio duty cycle is kept minimum to ensure low power operations.

**Fig. 8:** 47 nodes that form a network in a IoT-LAB testbed at Lille site

5.3.1 Results discussion

The graph in Fig. 9a shows the packet delivery ratio of all the nodes in the network. The average pdr for peer to peer data packet delivery in RPL is 78.64% which is less than the grid network shown in Fig. 7a. As all nodes have sufficient memory to hold routes, RPL’s delivery doesn’t fail due to lack of routes. It is due to multiple nodes using the same route for forwarding P2P packets. Though the path length is smaller in this topology, 73.8% of the data packets move through the root node indicating that the network has many DAG branches. This congestion reduces the delivery rate. However, C3P-RPL with the proposed centrality assisted P2P path selection method’s shorter and distributed routes improves the delivery to 90.14%. It proves that node betweenness centrality is a good measure to distribute simultaneous P2P traffic.

Fig. 9b displays the number of hops needed for a P2P packet to reach its destination. The average path length is 4.2 for RPL’s storing mode whereas it reduces to 2.9 for C3P-RPL with its out of DAG shortest paths. C3P-RPL’s method of utilizing topology information to derive shorter paths is more effective in reducing the length of a P2P packet’s travel. The outliers seen in this graph for the new network where nodes are added after the DAG creation represents the longer paths constructed in the DAG prior to the new node additions. There is no significant increase in overall packet hops which indicate that C3P-RPL is resilient for dynamic networks too.

The latency metric values shown in bar chart Fig. 9c, attests that the short and distributed P2P routes allows quicker turnaround time for a

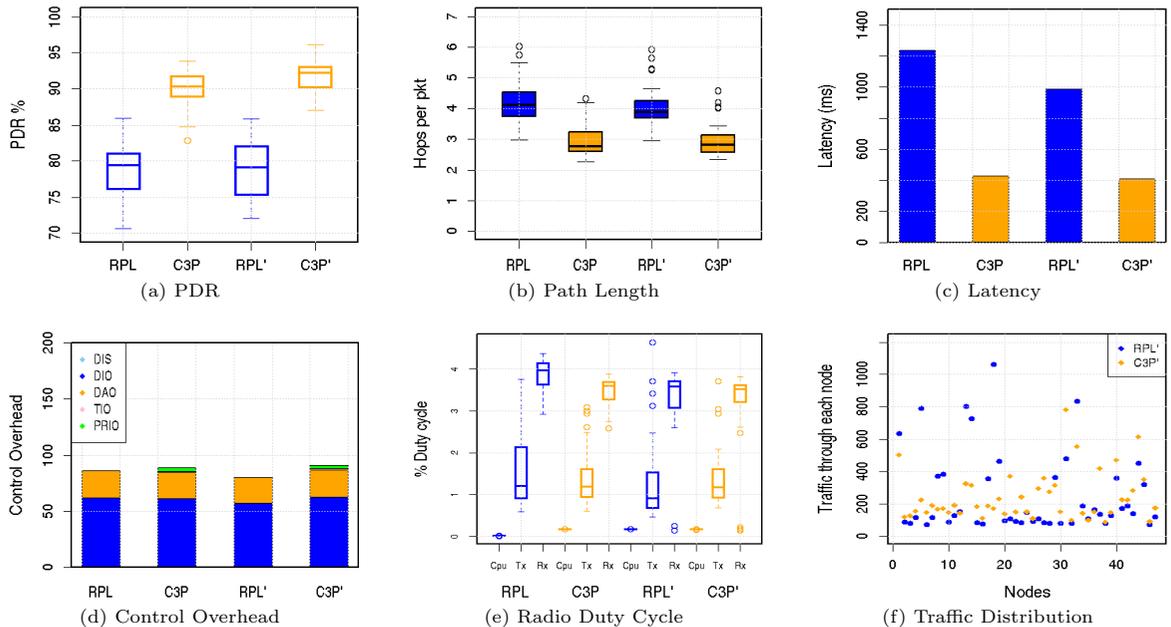


Fig. 9: Results for P2P traffic in a 47 node testbed at Lille (a) Packet delivery ratio (b) Path length of P2P routes (c) Packet Latency (d) Number of control packets transmitted (e) Percentage Radio Duty Cycle (f) Number of data packets passing through each node.

packet in the network along with better reliability. The average latency is brought down to 429.25 ms in C3P and 408.5 ms in C3P'. The proposed approach reduces the latency by around 65%. This is because the RPL nodes belonging to different DODAG branches go all the way to ancestor node and then down. In the tested network, the number of packets that go through the root node for C3P reduces by 73% than RPL. A lower latency is most suitable for devices to collaborate faster in real time.

As C3P-RPL introduces two new control messages and all nodes in the network are made to exchange them with the root node, a slight increase in control overhead is expected. The graph in Fig. 9d shows an increase in overhead for the proposed scheme. But it is very small when compared to the overall control overhead in the network. Unlike the grid topology, the control overhead of the proposed approach is similar to that of RPL. This is because of the shorter path length to root node when compared to the grid. So once the network stabilizes after getting the new routes, the overhead will not be increased further.

LLNs are known for low energy operations and the proposed implementation is measured against RPL in this respect. The nodes use ContikiMAC [44], its effective sleep-wakeup cycles allows the nodes to keep their active time for radio very less. Their radio can be in standby mode for the rest of the time. The graph in Fig. 9e shows that the duty cycle of C3P-RPL is better than that of RPL. This is corresponding to the reduction in the path length of the peer to peer routes. Also, the overall duty cycle is kept below 2% which ensures that the new approach is suitable for battery operated nodes. The receiver is relatively active than the transmitter because of the high number of neighbours in the testbed network than in the grid. On average, the grid network has only 3 neighbours whereas the network formed in the testbed has 8 neighbours.

Traffic distribution of the proposed approach can be understood from the graph in Fig. 9f. The extent of the distribution depends on the availability of the number of alternate nodes. Though the average number of neighbours is high in the testbed, few nodes have only couple of neighbours. Whereas all the nodes have similar number of

neighbours in grid. So the distribution is better in the grid network than in the testbed. With all these metrics, C3P-RPL with node centrality assisted P2P path selection shows reliable and faster peer to peer communication between all the nodes in the network.

6 Conclusion

Real-time device to device collaboration in a IoT network requires reliable and faster means of communication between them. RPL is the standard routing protocol that facilitates proactive routes between nodes and constructs a multi-hop network. Despite having sufficient memory to store routes, RPL's storing mode falls short in providing efficient point to point routes. This is due to the fact that RPL routes packets strictly along the DODAG and experiences congestion / collision around the branch heads. This paper proposed a new approach to select, store and sustain optimal peer to peer routes. Root node collates the topology of the network and a node betweenness centrality score selects an optimal single source shortest path tree that distributes the P2P data traffic. Local incremental updates to an existing shortest path tree sustains shortest routes in neighbouring nodes without having to disseminate the updates to all nodes. C3P-RPL is collaborative, proactive and decouples routing along DODAG to provide shorter and well distributed P2P paths. The simulations and testbed evaluations shows that the proposed approach improves overall P2P packet delivery, reduces path lengths of the routes, lowers delay in the network while ensuring low energy operations. This makes it suitable for battery operated nodes as well. In future, the work could be extended to scale up the operations through a clustering approach. Applications in IIoT, IoRT, IoMT and Smart cities could stand to gain as direct communication is essential for collaboration of smart services.

Declarations

Ethical approval and Consent to participate

This research work does not involve any human participants and / or animals.

Human and Animal Ethics

This research work does not involve any human participants and/or animals.

Consent for publication

All authors agree with the content and both of them has given their explicit consent to submit.

Availability of supporting data

The datasets are available with the authors and can be taken if required for the review process. The data would be published along with the the publication of the paper.

Competing interests

The authors have no relevant financial or non-financial interests to disclose.

Funding

No funds, grants, or other support was received for conducting this study.

Authors' contributions

T.Anusha: Conceptualization, Methodology, Software, Visualization, Validation and Writing manuscript. M.Pushpalatha: Methodology, Formal Analysis, Validation, Critical review and Supervision.

Acknowledgements

We would like offer our sincere gratitude to FIT IoT-LAB for their excellent testbed infrastructure and in-house tools for making it very convenient to use. We thank the team for the access and also for the prompt response to queries.

Authors' information

T Anusha has a B.E in Electronics and Communication Engineering and a M.E in Computer Science Engineering. She had worked in IT industry for a decade on various development projects. She is currently pursuing her Ph.D degree in SRM Institute of Science and Technology on IPv6 wireless mesh networks. Her research interests are Wireless Communication and Internet of Things.

M.Pushpalatha is Professor and Head for the Department of Computing Technologies, SRM Institute of Science and Technology. She was sponsored as a short term scholar to University of Southern California in 2011. She has received M.E degree from Madras University in 2001 and Ph.D degree in Computer Science and Engg. from SRM IST in 2014. On the research perspective, she continues to work as the principal investigator for many DRDO and DST projects and her research interests include Wireless ad hoc networks, Sensor Networks and Internet of Things.

References

- [1] Adegbija, T., Rogacs, A., Patel, C., Gordon-Ross, A., Microprocessor Optimizations for the Internet of Things: A Survey. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 37, 1, 7-20 (2018). <https://doi.org/10.1109/TCAD.2017.2717782>.
- [2] Neelakandan, S., Berlin, M.A., Tripathi, S. et al., IoT-based traffic prediction and traffic signal control system for smart city. *Soft Computing*, 25, 1224112248 (2021). <https://doi.org/10.1007/s00500-021-05896-x>
- [3] Xu, H., Yu, W., Griffith, D. and Golmie N., A Survey on Industrial Internet of Things: A Cyber-Physical Systems Perspective. *IEEE Access*, 6, 78238-78259 (2018). <http://doi.org/10.1109/ACCESS.2018.2884906>.
- [4] Wang, Y., Yan, H., Wan, J. (2016). Electronic Commerce Platform of Manufacturing Industry Under Industrial Internet of Things. In: Wan, J., Humar, I., Zhang, D. (eds) *Industrial IoT Technologies and Applications. Industrial IoT 2016. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol 173. Springer, Cham. https://doi.org/10.1007/978-3-319-44350-8_14
- [5] T. Winter et al., RFC 6550 - RPL: IPv6 routing protocol for low-power and lossy networks. Internet Eng. Task Force, Fremont, CA, USA, (2012). <https://doi.org/10.17487/RFC6550>.
- [6] Sobin C.C, A survey on Architecture, Protocols and Challenges in IoT. *Wireless Personal Comm.*, 112, 13831429 (2020). <https://doi.org/10.1007/s11277-020-07108-5>.
- [7] Porcu G. and Buron J. and Brandt A., Home Automation Routing Requirements in Low-Power and Lossy Networks. RFC 5826, Internet Eng. Task Force, (2010). <https://doi.org/10.17487/RFC5826>.
- [8] K. Pister, T. Phinney, P. Thubert and S. D. Pister and P. Thubert (Eds.), *Industrial Routing Requirements in Low-Power and Lossy Networks. RFC 5673*, Internet Eng. Task Force (2009). <https://doi.org/10.17487/RFC5673>.
- [9] T. Watteyne, T. Winter, D. Barthel and M. Dohler, (Eds.), *Routing Requirements for Urban Low-Power and Lossy Networks. RFC 5548* (2009). <https://doi.org/10.17487/RFC5548>.
- [10] M. Gupta and A. Conta (Eds.), *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. RFC 4443* (2006). <https://doi.org/10.17487/RFC4443>.
- [11] P. Levis, T. Clausen, J. Hui, O. Gnawali and J. Ko, *The Trickle Algorithm. Internet Eng. Task Force, RFC 6206*, (2011). <https://doi.org/10.17487/RFC6206>.
- [12] B. Peres, B. P. Santos, O. A. de O. Souza, O. Goussevskaia, M. A. M. Vieira, L. F. M. Vieira and A. A. F. Loureiro, *Matrix: Multihop Address allocation and dynamic any-To-any Routing for 6LoWPAN. Computer Networks*, 140, 28-40 (2018). <https://doi.org/10.1016/j.comnet.2018.04.017>.
- [13] T. Clausen, U. Herberg, and M. Philipp, A critical evaluation of the IPv6 routing protocol for low power and lossy networks (RPL). In: *Proc. IEEE Int. Conf. Wireless Mobile Comput. Netw. Commun. (WiMob)*, Wuhan, China, 365372 (2011). <https://doi.org/10.1109/WIMOB.2011.6085374>.

- [14] H. Kim, J. Ko, D. E. Culler and J. Paek, Challenging the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL): A Survey. *IEEE Comm. Surveys & Tutorials*, 19, 4, 2502-2525 (2017). <https://doi.org/10.1109/COMST.2017.2751617>.
- [15] M. Goyal (Ed.), E. Baccelli, M. Philipp, A. Brandt and J. Martocci, Reactive Discovery of PointtoPoint Routes in Low-Power and Lossy Networks. RFC 6997 (2013). <https://doi.org/10.17487/RFC6997>.
- [16] Y. Kim and J. Paek, NG-RPL for Efficient P2P Routing in Low-Power Multi-hop Wireless Networks. *IEEE Access*, vol. 8, pp. 182591 - 182599 (2020).
- [17] M. Mahyoub, A.S H. Mahmoud, M. Abu-Amara and T.R. Sheltami, An Efficient RPL-Based Mechanism for Node-to-Node Communications in IoT. *IEEE Internet of Things Journal*, 8, 9, 7152-7169 (2021). <https://doi.org/10.1109/JIOT.2020.3038696>.
- [18] B. Ghaleb, A. Y. Al-Dubai, E. Ekonomou, A. Alsarhan, Y. Nasse, L. M. Mackenzie and A. Boukerche, A Survey of Limitations and Enhancements of the IPv6 Routing Protocol for Low-Power and Lossy Networks: A Focus on Core Operation. *IEEE Comm. Surveys & Tutorials*, 21, 2, 1605-1635 (2019). <https://doi.org/10.1109/COMST.2018.2874356>.
- [19] P. O. Kamgueu, E. Nataf and T. D. Ndie, Survey on RPL enhancements: A focus on topology, security and mobility. *Computer Communications*, 120, 1021 (2018). <https://doi.org/10.1016/j.comcom.2018.02.011>.
- [20] M. Zhao, A. Kumar, T. Ristaniemi, and P. H. J. Chong, Machine-to-Machine Communication and Research Challenges: A Survey. *Wireless Pers. Commun.*, 97, 35693585 (2017). <https://doi.org/10.1007/s11277-017-4686-1>.
- [21] Z. Wang, L. Zhang, Z. Zheng and J. Wang, Energy balancing RPL protocol with multipath for wireless sensor networks. *Peer-to-Peer Netw. Appl.*, 11, 10851100, (2018). <https://doi.org/10.1007/s12083-017-0585-1>.
- [22] Nesary Moghadam, M., Taheri, H. and Karari, M., Minimum cost load balanced multipath routing protocol for low power and lossy networks. *Wireless Netw* 20, 2469-2479 (2014). <https://doi.org/10.1007/s11276-014-0753-7>.
- [23] Pushpa Mettilsha, J., Sandhya, M.K. and Murugan K, RPR: Reliable path routing protocol to mitigate congestion in critical IoT applications. *Wireless Networks*, 27, 5229-5243 (2021). <https://doi.org/10.1007/s11276-021-02805-w>.
- [24] H. Kim, J. Paek and S. Bahk, QU-RPL: Queue Utilization based RPL for Load Balancing in Large Scale Industrial Applications. 2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), 265-273 (2015). <https://doi.org/10.1109/SAHCN.2015.7338325>.
- [25] E. Baccelli, M. Philipp, and M. Goyal, The P2P-RPL routing protocol for IPv6 sensor networks: Testbed experiments. In: *Proc. Int. Conf. Softw. Telecommun. Comput. Netw. (SoftCOM)*, Split, Croatia, 16, (2011).
- [26] Zhao, M., Kumar, A., Joo Chong, P.H. et al. A comprehensive study of RPL and P2P-RPL routing protocols: Implementation, challenges and opportunities. *Peer-to-Peer Netw. Appl.* 10, 12321256 (2017). <https://doi.org/10.1007/s12083-016-0475-y>.
- [27] J. Ko, J. Jeong, J. Park, J. A. Jun and J. Paek, DualMOP-RPL: Supporting Multiple Modes of Downward Routing in a Single RPL Network. *ACM Transactions on Sensor Networks*, 11, 2, Article. 39, 1-20 (2015). <https://doi.org/10.1145/2700261>.
- [28] X. Zhong and Y. Liang, Scalable Downward Routing for Wireless Sensor Networks Actuation. *IEEE Sensors Journal*, 19, 20, 9552-9560 (2019). <https://doi.org/10.1109/JSEN.2019.2924153>.
- [29] M. Zhao, I. W. Ho and P. H. J. Chong, An Energy-Efficient Region-Based RPL Routing Protocol for Low-Power and Lossy Networks. *IEEE Internet of Things Journal*, 3, 6, 1319-1333 (2016).

<https://doi.org/10.1109/JIOT.2016.2593438>.

- [30] NS3 - Open source network simulator for networking research. <https://www.nsnam.org/> (2022). Accessed 01 Mar. 2022.
- [31] M. O. Farooq and D. Pesch, ERPL: An Enhanced Peer-to-Peer Routing Mechanism for Low-Power and Lossy Networks. 11th IFIP Wireless and Mobile Networking Conference (WMNC), 1-8 (2018). <https://doi.org/10.23919/WMNC.2018.8480918>.
- [32] W. Gan, Z. Shi, C. Zhang, L. Sun and D. Ionescu, MERPL:A More Memory-efficient Storing Mode in RPL. 2013 19th IEEE International Conference on Networks (ICON), 1-5 (2013). <https://doi.org/10.1109/ICON.2013.6781985>.
- [33] C. Kiraly, T. Istomin, O. Iova and G. P. Picco, D-RPL: Overcoming memory limitations in RPL point-to-multipoint routing. 2015 IEEE 40th Conference on Local Computer Networks (LCN), 157-160 (2015). <https://doi.org/10.1109/LCN.2015.7366295>.
- [34] E. Rojas, H. Hosseini, C. Gomez, D. Carrascal and J. R. Cotrim, Outperforming RPL with scalable routing based on meaningful MAC addressing. *Ad Hoc Networks*, 114, 1570-8705 (2021). <https://doi.org/10.1016/j.adhoc.2021.102433>.
- [35] Skiena, S. *The Algorithm Design Manual*, 2nd ed., Springer-Verlag (2010). <https://doi.org/10.1007/978-1-84800-070-4>.
- [36] G. Ramalingam and T. W. Reps, On the computational complexity of dynamic graph problems. *Theor. Comput. Sci.*, 158 (1 and 2), 233-277 (1996).
- [37] A. Slobbe, E. Bergamini and H. Meyerhenke, Faster Incremental All-pairs Shortest Paths (2016). <https://doi.org/10.13140/RG.2.1.3692.2005>.
- [38] Texas Instruments, MSP430F5 series Mixed-Signal Micro controllers. <https://www.ti.com/lit/gpn/msp430f5437> (2022). Accessed 01 March 2022.
- [39] Texas Instruments, CC2520 2.5 GHz IEEE 802.15.4 RF Transceiver. <https://www.ti.com/lit/gpn/cc2520> (2022). Accessed 01 March 2022.
- [40] Contiki Operating System. <https://github.com/contiki-os/contiki> (2022). Accessed 01 Mar 2022.
- [41] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele and T. Watteyne, FIT IoT-LAB: A Large Scale Open Experimental IoT Testbed. *IEEE World Forum on Internet of Things (IEEE WF-IoT)*, 459-464, (2015). <https://doi.org/10.1109/WF-IoT.2015.7389098>.
- [42] AT86RF231, a low power 2.4 GHz transceiver designed for industrial and consumer IEEE 802.15.4, ZigBee, RF4CE, SP100, WirelessHART, ISM, and high data rate applications. <https://www.microchip.com/en-us/product/AT86RF231>. Accessed 01 Mar 2022.
- [43] Arm Cortex-M3 MCU with 512 Kbytes of Flash memory, 72 MHz CPU, motor control, USB and CAN. <https://www.st.com/en/microcontrollers-microprocessors/stm32f103re.html>. Accessed 01 March 2022.
- [44] A. Dunkels, *The ContikiMAC Radio Duty Cycling Protocol* (2011). SICS Technical Report, T2011:13, ISSN: 1100-3154.

the Conclusions and Referenc