Journal of Network and Computer Applications 164 (2020) 102675

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

A robust multicast communication protocol for Low power and Lossy networks

Mauro Conti [a], Pallavi Kaliyar [a], Chhagan Lal [b,*]

[a] Department of Mathematics, University of Padova, Italy
[b] Simula Research Laboratory, Norway

ARTICLE INFO

Index Terms:
Internet of Things
RPL
6LoWPAN
IPv6
Multicast routing
LLNs
Communication security

ABSTRACT

Routing Protocol for low power and Lossy networks (RPL) is a standardized routing protocol for low power and lossy networks (LLNs) such as the Internet of Things (IoT). RPL was designed to be a simple (but efficient) and practical networking protocol to perform routing in IoT networks that consists of resource constrained devices. These tiny intercommunicating devices are currently in use in a large array of IoT application services (e.g., eHealth, smart agriculture, smart grids, and home automation). However, the lack of scalability and the low data communication reliability due to faulty links or malicious nodes, still remains significant challenges in the broader adoption of RPL in LLNs. In this paper, we propose *RECOUP*, a robust multicast communication routing protocol for Low power and Lossy Networks. RECOUP efficiently uses a low-overhead cluster-based multicast routing technique on top of the RPL protocol. RECOUP increases the probability of message delivery to the intended destination(s), irrespective of the network size and faults (such as broken links or non-responsive nodes), and in the presence of misbehaving nodes. An implementation of RECOUP is realized in Contiki. Our results show the effectiveness of RECOUP over state-of-art protocols concerning packet delivery ratio to 25%, end-to-end delay down to 100 ms, and low radio transmissions required for per packet delivery to 6 mJ. Moreover, it minimizes the impact of various topologies (i.e., rank and sybil) and data communication (i.e., blackhole, wormhole, and jamming) attacks that targets an IoT networking infrastructure.

1. Introduction

In Internet of Thing (IoT) networks, the sensors collect data and send it to base stations or actuators for storage, processing, and service creation (Al-Fuqaha et al., 2015). The IoT devices are usually clubbed together, logically, into groups based on their functionalities and utility. In particular, an IoT network consists of constrained sensor devices (also called *motes*) that create a Low-power Wireless Personal Area Network (LoWPAN) in which communication is done using a compressed Internet Protocol Version 6 (IPv6). LoWPAN over IPv6 (i.e., 6LoWPAN) uses the IEEE 802.15.4 as the data-link and physical layer protocol (Kushalnagar et al., 2007; ShelbyZach et al., 2010).

For routing in resource constrained networks such as the IoT, the RPL (Winter et al., 2012) is considered as an idle routing solution (Kim et al., 2017a; Raoof et al., 2019). The RPL mainly supports point-to-point (P2P) communications (i.e., unicast), but it also provides an optional support for multicast routing as well. The use of RPL's unicast routing exhibits several issues, such as low scalability, high propagation delay, and high energy consumption becomes significant routing issues (Hui et al., 2012). Therefore, recently the researchers are exploring the multicast routing options in RPL. However, in RPL, the availability of a single route between source and destination pairs is a key issue that adversely effects the communication reliability and security in the underlying network. It is because a single broken link or non-responsive node due to some network fault or a compromised node that resides on the route could disrupt the routing process (Kim et al., 2017b). Therefore, our work aims to improve the data communication in IoT networks, mainly in terms of routing robustness (i.e., fault tolerance, low delay, and high network throughput), scalability, and security.

1.1. Contributions

In this paper, we propose a robust (i.e., able to cope with link and node failures, and minimize the impact of security attacks) group communication protocol namely RECOUP, for efficient data communica-

* Corresponding author.
E-mail addresses: conti@math.unipd.it (M. Conti), pallavi@math.unipd.it (P. Kaliyar), chhagan@math.unipd.it (C. Lal), chhagan.iiita@gmail.com (C. Lal).

https://doi.org/10.1016/j.jnca.2020.102675
Received 10 April 2019; Received in revised form 15 January 2020; Accepted 21 April 2020
Available online XXX
1084-8045/

tion in LLNs such as the IoT. The key functionalities of RECOUP are as follow: (i) virtual clusters creation on top of RPL's logical Destination Oriented Directed Acyclic Graph (DODAG) topology, (ii) perform upward and downward multicast routing in DODAG by using RPL's storing mode of operation (also called as MOP3), and (iii) optimized inter-cluster routing for quick dissemination and high delivery rate of multicast packets. These functionalities of RECOUP leads to low data packet propagation delay, high packet delivery ratio, and minimal effect of various topology and communication attacks in the network.

In summary, the key contributions of this work are as follow.

- We design and fully implement *RECOUP*, a novel robust multicast routing protocol for improved data communication in IoT networks. RECOUP makes efficient use of RPL's control messages to implement its optimized cluster based horizontal routing mechanism, thus, it avoids additional memory and control overhead in its execution process. Our detailed discussion on RECOUP's data communication reliability and robustness in resisting an array of security threats in different IoT networking scenarios show the major advantages of RECOUP. The paper also report the key implementation issues of RECOUP which includes energy consumption and memory requirements on network nodes while running the RECOUP protocol.
- We perform a comprehensive performance evaluation of RECOUP concerning various network metrics such as end-to-end delay, packet delivery ratio, average path cost, energy consumption, and memory requirements. To shows the efficiency of RECOUP regarding communication robustness (i.e., broken links or faulty nodes) and resistance to security threats (i.e., in presence of an adversary), the result evaluation is done with varying network size and in the presence of attacker nodes in the target scenario. Furthermore, to show its effectiveness, we compare RECOUP with the following RPL based state-of-the-art multicast routing protocols: (i) ESMRF (Fadeel et al., 2015), an enhanced stateless multicast RPL-based forwarding protocol, and (ii) BMRF (Lorente et al., 2017), a bidirectional multicast RPL forwarding protocol. The implementation is done in *Cooja*, the Contiki network emulator (Romdhani et al., 2016), which is widely used for deploying energy-constrained and memory-efficient LLNs. We make available[1] an open-source implementation of RECOUP along with all the source code to the research community for future research in this direction.

1.2. Organization

The rest of this paper is organized as follow. In Section 2, we discuss the state-of-the-art IoT routing protocols and techniques that addresses the data communication issues in IoT networks. In Section 4, we first present the system and adversary model, and then the design and implementation details of RECOUP along with its working methodologies. In Section 5.1, we present the detailed performance evaluation of RECOUP concerning various metrics using the *Contiki Cooja* emulator. Finally, Section 6 concludes our work.

2. Background and related work

In this section, first we present a brief overview of the RPL and its extension routing protocols that are proposed for 6LoWPAN networks. Then we discuss the related work concerning secure and efficient data communication in RPL-based IoT networks.

2.1. Routing protocol for low power and lossy networks (RPL)

RPL (Winter et al., 2012) creates a virtual routing topology called Destination-Oriented Directed Acyclic Graph (DODAG) on top of the underlying random physical topology. DODAG is a directed graph with no loops, oriented towards a root node (e.g., a LLN/border router). In DODAG, each node by default have multiple parents towards the root, however, only a preferred one which is selected based on routing metric and objective function (OF) is used for forwarding data packets, while the others are kept as backup routes. The structure of DODAG naturally supports multipoint-to-point communication in RPL, which provides communication from the nodes to the root with minimal routing states. The DODAG topology is created and maintained via ICMPv6 control packets known as DODAG Information Objects (DIO). Each node in RPL advertises DIO messages, which contains the link and node metrics (e.g., expected transmission count (ETX), residual energy) and an OF that are used by each node to select its preferred parent. Moreover, when a node receives a DIO message it calculate its own rank (i.e., $R_i = R_p + 1$, where R_p is the parent rank). Once a node select its preferred parent, then it notifies the parent by sending a DAO message, and the parent confirm it by replying with a DAO-ACK message. To maintain the DODAG, DIO packets are rebroadcast by each node based on the Trickle algorithm (Levis et al., 2011), which is an adaptive technique that tries to achieve a balance between reactivity to topology changes (fast convergence/recovery) and control overhead (energy consumption). In particular, Trickle ensures that DIO packets are rebroadcast at slow pace when the network is stable, and aggressively when it is unstable. DIO packets are also transmitted upon request when a node receives a DODAG information solicitation (DIS) packet, which could be sent by a new node that wants to join the DODAG.

Apart from multipoint-to-point communication, the RPL supports point-to-multipoint and point-to-point communications in two modes called storing and non-storing modes. In RPL's storing mode (table-driven routing), the non-root nodes store the routing information about all their descendant nodes, while in non-storing mode (source routing) the routing information about all the nodes is stored at the root. In both the modes, the routing information is collected using Destination Advertisement Object (DAO) control packets, which are transmitted by each node in the network to announce itself as a possible destination to the root. DOA packets are propagated towards the root, via a parent, therefore establishing "downwards" routes along the way. The detailed working of RPL and its features are out of the scope of this paper. Therefore, we direct the interested readers to more comprehensive literature given in Winter et al. (2012) and Kim et al. (2017a).

2.2. RPL extensions for secure and efficient data communications

The first extension that uses the RPL functionality is proposed in Oikonomou et al. (2013), and it is called Stateless Multicast RPL Forwarding (SMRF). In SMRF, nodes only process the multicast packets which are coming from their preferred parents, thus, SMRF only allows the forwarding of multicast packets in downward direction in the RPL DODAG tree. To improve the functionality of SMRF, following two extensions are proposed: (i) Enhanced Stateless Multicast RPL Forwarding (ESMRF) (Fadeel et al., 2015), in which sources of multicast traffic encapsulates their multicast packet in an ICMPv6 delegation packet and send it to the root for forwarding, and (ii) Bi-Directional Multicast Forwarding Algorithm (BMFA) (Papadopoulos et al., 2017), which improves SMRF and enable multicasting in upward and downward directions. Finally, authors in Lorente et al. (2017) propose the *Bidirectional Multicast RPL Forwarding* (BMRF) protocol, which fully utilizes the potential of RPL's non-storing mode to overcome various disadvantages of SMRF and its extensions. In BMRF, when a node wants to send a multicast message, it performs the bidirectional forwarding. BMRF provides a choice for Link Layer unicast, broadcast, or mixed mode to forward a multicast packet at a parent node. Link Layer unicast or broadcast depends upon the number of interested children and mix mode depends upon whether the number of interested children are larger than a pre-defined threshold value. In addition, BMRF added one more new feature that allows a node to un-subscribe itself from a multi-

[1] https://github.com/pallavikaliyar/RECOUP.

cast group by sending a DAO message to the preferred parent. The main advantages of BMRF includes that it avoids duplicates and overheads, there is no delivery disorder, and it enables multi-sourcing, i.e., at a single time in a network more than one source node can send multicast messages to the same multicast destination address. However, the BMRF also possess a set of disadvantages such as higher energy consumption, latency, and lower communication reliability and security.

As RPL and its extensions are the most used routing protocols in IoT networks. We now briefly discuss the security challenges that these protocols faces during their routing process. Authors in Pu and Hajjar (2018) investigates the forwarding misbehaviour (i.e., selective packet discarding) and propose a countermeasure in LLNs that runs the RPL protocol. The basic idea is to monitor the forwarding (mis)behaviour of each node to observe the packet loss rate, and then compare the packet loss rate of the parent node with the neighbour nodes. To ensure that the packet loss is due to misbehaviour and not due to bad channel quality, the nodes use one time retransmission techniques. Similarly, using the monitoring information of the nodes about the data packets forwarding, a trust-based intrusion detection system based on RPL is presented in Medjek et al. (2017), to countermeasure mobile sybil attacks in IoT networks.

In Ahmed and Ko (2016), authors propose a solution to mitigate blackhole attack in RPL by using a mechanism similar to *watchdog*, in which the neighbour nodes keep record of a nodes' activities and analyze it to find any malicious behaviour. Authors in Dvir et al. (2011) and Perrey et al. (2016) addresses the *rank*[2] attack, which is an attack specific to RPL. VeRA (Dvir et al., 2011) effectively fixes the vulnerabilities caused by the false rank of a node and the DODAG version number dissemination. VeRA does it by adding reverse hash chaining to DIO messages due to which receivers shall be enabled to verify the advertised hierarchy. However, in Perrey et al. (2016) authors show that VeRA remains vulnerable to rank attacks by forgery and replay, and they propose TRAIL (Trust Anchor Interconnection Loop), which aims to discover and isolate bogus nodes. The key idea is to validate upward paths to the root using a round trip message. This is achieved without relying on encryption chains (as in VeRA). In TRAIL, a node can conclude rank integrity from a recursively intact upward path. Recently, authors in Conti et al. (2018) propose a secure and scalable RPL routing protocol (SPLIT) for IoT networks. SPLIT uses a lightweight remote attestation technique to ensure software integrity of network nodes, which ensures their correct behaviour. To avoid additional overhead caused by attestation messages, SPLIT piggybacks attestation process on the RPL's control messages.

3. System and adversary models

In this section, we present the details of the system and adversary models on which RECOUP is implemented and evaluated.

3.1. System model

In our work, we assume that the system model has the following properties.

- The target network consists of a set $D = \{D_1, D_2, \ldots D_n\}$ of size n resource constraint IoT nodes (i.e., sensors and actuators). These nodes are static within the given IoT network area. We consider that all the nodes are homogeneous in terms of resources, but could be different in terms of their functionalities depending upon the configured sensor type such as temperature, illumination, audio, pressure, to name a few. Nodes with similar functionalities are grouped together to form a multicast group in the network. All the nodes are configured using the standard layered protocol stack of IoT. At network layer the nodes use RPL MOP3 (i.e., storing with multicast support) over IPv6 as a routing protocol for data communication.
- At the start, n nodes are deployed in a random fashion, and the RPL creates a virtual DODAG on top of the physical network topology. Apart from these n nodes, the network also has resourceful nodes called LLN border router (LBR) which acts as the root for the DODAGs in the network. A network could have more than one DODAG represented by different DODAG IDs (DID_i) and different root nodes. Each node in the DODAG has a *rank* value which specify its level in DODAG, i.e., distance from the root. The rank of the root is set to 0, and the rank associated with a node increases with its distance from root.
- In RECOUP, each DODAG is divided into a set of clusters, and the nodes having rank 1 will act as the clusterheads. For instance, the nodes with IDs 1, 5, 7, 12, 14, 15, and 16 in Fig. 1 will act as clusterheads within that particular DODAG. Each cluster is represented by a unique ID (*CID*). It can be seen from Fig. 1 that nodes in DODAG are arranged in a parent and child structure, each parent store information about its children which includes their subscription for a multicast group among other data. A node could be subscribed for more than one multicast group depending upon the usage requirements of IoT application running on top of the network.
- Multicast routing is used to send data messages to a group of nodes with similar functionalities. However, the network also supports point-to-point and multipoint-to-point routing. The source of the multicast message could be the root node or a member of a multicast group. Data exchanged between two communicating nodes that are not within each others radio range will be forwarded by intermediate nodes.
- RECOUP uses the following additional or enhanced data structures at different nodes in the DODAG.
  - *Neighbour Table (N_{tab}):* The N_{tab} can be simply created by extending the functionalities of the RPL's routing table that a node stores for routing in RPL's storing mode. In RECOUP, each N_{tab} entry stores the following information about a neighbour node (say N): (i) cluster ID (*CID*) of $N's$ cluster, (ii) node ID of N (*NID*), and (iii) rank of N. As stated before that we implement N_{tab} on top of the existing *information/routing table* that already exists at all the nodes in the network. The additional information that RECOUP adds is the *CID* of a node and the entries for the neighbour nodes that are not the descendants. The N_{tab} is associated with a timer called *TrickleTimer* (*TT*), and the nodes update the N_{tab} with new network information once this timer expires. The value of the timer is set by the network administrator depending upon the RPL DODAG reformation i.e., if any new node joins the existing DODAG or any existing node changes its parent node within the DODAG.
  - *Duplicate Detection Table (DD_{tab}):* RECOUP uses DD_{tab} at root node: (i) to check for duplicates because in RECOUP the same packet is travelling towards root through multiple clusters, and (ii) to hold the received packet for a variable time duration while waiting for all the duplicates to receive from multiple clusters. The DD_{tab} consists of a set of entries, where each entry has the following information about the received multicast packet (MP_i): (i) source address (S_{id}), (ii) destination address (D_{id}), (iii) set of cluster ID(s) from which MP_i is received (CID_{set}) so far, i.e., ID of the cluster from which the packet or its duplicates has been received, (iv) packet sequence number (N_{seq}), (v) forwarding timer (F_{time}), and (vi) a buffer to hold the MP_i until the associated F_{time} expires. Upon expiration of the F_{time}, the packet is processed and the entry is removed from the DD_{tab}. The tuple $< S_{id}, D_{id}, N_{seq} >$ is used to discard duplicates. We have implemented DD_{tab} as a dynamic link list at LBRs or root only. Additionally, same as the traditional

[2] An attacker decreases its rank to spoil the routing topology and attract traffic from neighbour nodes, which degrades packet delivery performance when combined with blackhole, wormhole, or selective forwarding attacks.

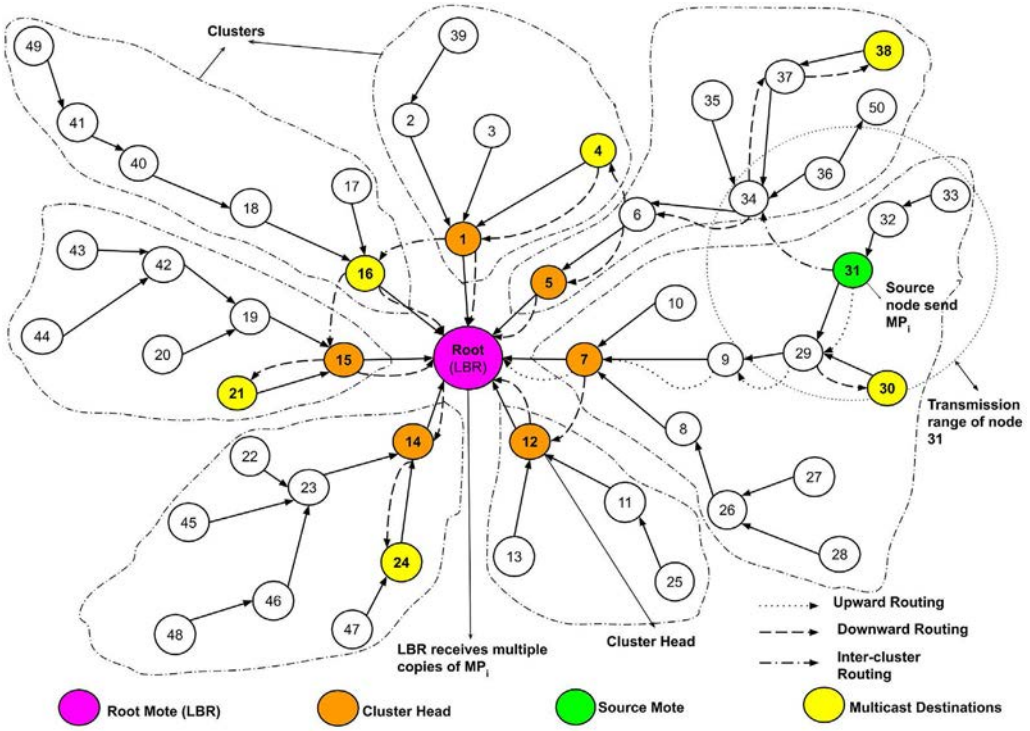

Fig. 1. Example: RECOUP cluster formation and forwarding mechanism.

RPL, each node in RECOUP also stores the DD_{tab} whose function is limited to just detect and discard the duplicate packets, i.e., the entries in DD_{tab} at non-root nodes only consists of the tuple $< S_{id}, D_{id}, N_{seq} >$.

3.2. Adversary model

The use of IoT networks in a large array of user-centric applications make these networks a high profit target for adversaries. Hence, the adversaries would try their best to equip themselves with advanced equipment, which means they would have few technical advantages over the IoT nodes. In our target IoT network, an attacker is interested in minimizing the connectivity of the network to prevent the LBR or members of a multicast group from detecting important events, thus impairing their decision making system. To achieve this goal, the attacker selects at each time t a node to compromise from the set D. In fact, the attacker chooses the node which maximizes the adverse impact on the IoT services running on top of the networking infrastructure.

In our target IoT network, the adversaries are assumed to have the following characteristics:

- The adversary is resourceful, and it could perform the rank, jamming, blackhole, eavesdropping, and wormhole attacks. To launch the aforementioned attacks, it can compromise an existing node or it can be part of an existing network as a new node. However, we assume that the adversary cannot compromise the LBR (i.e., DODAG root).
- The adversary will not interfere with the proper functioning of the network such as modifying the data packets, generating new messages, destroying network devices, and tempering with the key distribution and management operations. It is because such activities can be easily detected by an IDS and could put the adversary at risk of being caught (Zarpelo et al., 2017).

4. Our proposal: RECOUP

In this section, we discuss the working methodology of RECOUP along with its design considerations, characteristics, and optimized routing process.

4.1. Design considerations

Below is the list of design considerations along with their functioning details that were taken into account while designing the RECOUP protocol.

- *Cluster Formation:* In RECOUP, the clusters are created virtually on top of RPL's DODAG tree. We limit the rank for clusterheads to 1, i.e., only the children of the root/LBR can act as clusterheads (please refer to Fig. 1). Thus, the number of clusters in a network are equal to the number of children of DODAG root, and no clusterhead selection algorithm is needed since only the children of root will become the clusterhead.
- *Cluster ID:* The DODAG root will assign a unique identification to its clusters called as cluster ID (CID). All the nodes that belongs to the same cluster will share a common CID.
- *Inter-cluster routing:* RECOUP uses inter-cluster routing, in which a node in a cluster could forward data packets to its neighbouring cluster nodes.
- *Information Storage:* The nodes in RECOUP will be configured with RPL MOP3 mode, therefore, each node will store the essential information needed to route the messages in upward and downward routes in their N_{tab}. Additionally, the nodes will also store the duplicate detection table DD_{tab} and N_{tab} as described in Section 3.1.
- *Duplicate Avoidance:* In RECOUP, to minimize the duplicate messages the following optimization's are included: (i) the inter-cluster forwarding is limited by using a threshold hop-count value, (ii) during inter-cluster routing, a node will send a message to only one neighbour from the group of neighbours if this all neighbours belong to the same cluster, (iii) we use low transmission range (i.e., 25 m) for data communication to

Table 1
Symbol table.

Symbol	Meaning
X	node in DODAG
P	parent of a node
TX_{MP}	link layer transmission time of a multicast packet
X_r	rank of node X
MP_i	ith multicast data packet
Src (MP_i)	source of MP_i
MG_i	ith multicast group in network
X_{CID}	ID of X's cluster
IC	set of interested children of a node
DD_{tab}	duplicate detection table
F_{time}	MP_i hold time at LBR
CID_{visit}	set of cluster IDs at LBR from which MP_i is received
$C2C$	inter cluster routing set/reset flag
X (N_{tab})	neighbour table at X
Pkt_{drop}	maximum hop-count for MP_i in inter-cluster forwarding
NC_{ID}	ID of ith neighbour cluster
N_j	jth neighbour from a NC
C_{travel}	set of clusters travelled by MP_i via inter-cluster forwarding

reduce both, the overlapping in neighbouring clusters and the re-transmissions required to forward a message to its next hop, and (iv) a node will not forward a data packet in the direction from which it has been received. The details about the aforementioned duplicate avoidance steps will be presented in Section 4.2.

- *Multi-directional Forwarding:* The multicast packet will be forwarded in all directions, which includes upwards (i.e., preferred parent), downwards (i.e., interested children) and neighbour (i.e., inter-cluster routing) nodes. This feature of RECOUP plays an important role, specifically to decrease the propagation delay and to the increase network scalability, fault tolerance, and robustness against security threats.

4.2. Approach description

In this section, we present the working methodology of *RECOUP* protocol, which is mainly divided into two phases. The first phase consists of the DODAG and cluster formation, and the second phase consists of the cluster-based data packet routing technique. The symbols used for explaining the routing mechanism of RECOUP are shown in Table 1.

4.2.1. DODAG and cluster creation in RECOUP

The cluster formation is done in parallel with the DODAG creation by using the following steps.

- In RECOUP, a node with rank 1 will act as clusterhead, and any node that joins a cluster will use the same CID that is assigned to the clusterhead. Initially, the clusterheads receive their CID from the LBR in RPL's DAO-ACK control messages.
- Once the clusterheads have their unique CID, they broadcast DIO messages with their ID. The nodes that receive these DIO messages will select a preferred parent, keep the received cluster ID, calculate their own rank by increasing the parent rank by 1, and then broadcast the DIO again in the network. This process is repeated until the network constructs the routing topology (i.e., DODAG) and along with it the cluster formation will also be done.

By executing the aforementioned steps, RECOUP creates the required clusters in parallel with the formation of the DODAG, thus minimizes both, the control overhead messages and the network convergence time. As the cluster formation is closely coupled with the RPL's DODAG creation, there is no need to perform the cluster maintenance as it happens automatically with DODAG's re-creation process.

Algorithm 1 LBR multicast packet routing process in RECOUP.

INPUT at a Node: Data packet
OUTPUT: Forward the data packet towards its destination
1: **if** X has a MP_i to send to MG_i **then**
2: **if** $X \in MG_i$ **then**
3: deliver MP_i up to the network stack
4: **end if**
5: **if** Src (MP_i) = LBR **then**
6: perform only downward routing
7: LBR use source-routing to route MP_i
8: **else**
9: **if** $X = LBR$ **then**
10: create new entry in DD_{tab} for MP_i
11: $F_{time} \leftarrow (TX_{MP} \times X_r + \alpha)$ and associate F_{time} with the entry
12: **while** $F_{time} \neq 0$ **do**
13: update CID_{visit} for each duplicate MP_i
14: **end while**
15: **if** ($IC \leftarrow IC \cap CID_{visit}$) = NULL **then**
16: drop the packet
17: **else**
18: set $C2C \leftarrow 1$
19: transmit MP_i to members of IC
20: GOTO Algorithm 2
21: **end if**
22: **else**
23: GOTO Algorithm 2
24: **end if**
25: **end if**
26: **end if**

4.2.2. Data routing in RECOUP

Due to the use of multicast routing in a large array of practical IoT applications, we evaluate and analyze the performance of RECOUP mainly for multicast communications. However, RECOUP also supports unicast routing. When a source node wants to send a multicast packet using RECOUP, it transmits the packet in following three directions: (i) upward, i.e., towards LBR through its preferred parent; (ii) downward, i.e., towards interested children who are registered for the multicast group that is specified in the destination address of transmitting packet header; and (iii) inter-cluster, i.e., toward neighbour(s) with different cluster ID. In case where the source node has multiple neighbours that belongs to the same cluster, the packet is sent to only one of the neighbour from that cluster. It is because if a single node in the cluster receives the packet, later it will be disseminated in the whole cluster. Next, we discuss the functionality of the routing mechanism of RECOUP for all possible data communication scenarios in an IoT network.

Routing at LBR Node. Algorithm 1 shows the routing procedure at LBR/root node. When LBR has a multicast data packet (say MP_i) to send, it checks if it is the source or intermediate hop of MP_i. If LBR is the source of MP_i, then it performs the downward multicast routing by simply doing the source routing, which uses the global network information stored in its routing table. In particular, the LBR send MP_i to its interested children, i.e., the children that are subscribed to the destination multicast address specified in the MP_i. The children also do the same, and the process continues until the packet reaches to all the subscribed nodes of the multicast group. On the other hand, if the LBR is not the source of MP_i, it indicates that the packet is received by root from one or more of the underlying clusters. As the same packet might be travelling towards LBR from multiple clusters due to our inter-cluster routing, the LBR will possibly receive duplicate copies of MP_i. When the LBR receive the first copy of MP_i, it creates a new entry in its DD_{tab}. The entry contains a buffer to store the received packet along

with other information as described in Section 3.1. Additionally, the LBR associates a timer called F_{time} with each new entry. The value of F_{time} is calculated by multiplying the rank of the source of MP_i to the time taken to transmit a packet from one hop to the next hop. A random time value (say α) is also added to F_{time} to ensure that the LBR will receive all the duplicates of MP_i from the clusters. The lower value of α will increase the number of duplicates in the network because the LBR might falsely forward the MP_i in the cluster(s) which already have the MP_i through inter-cluster routing. Alternatively, the large value of α will increase the waiting time of MP_i at LBR, which will increase the routing end-to-end delay in network. In RECOUP, the initial value of α is set to 0, and it is gradually increased or decreased in proportional to the number of duplicates received for an MP_i after it is forwarded by the LBR. Specifically, for a data session, apart from the initial value of α which is set to 0, the subsequent values of α are estimated as follows.

$$\alpha = \alpha_{prev} + (T_{X_{MP}^{last}} - F_{time}^{prev}) \quad (1)$$

Where α_{prev} is the previous value of α, $T_{X_{MP}^{last}}$ is the total time by which all the copies of MP_i has been received at LBR, and F_{time}^{prev} is the previous hold time at LBR for MP_i.

Once the F_{time} associated with an entry in DD_{tab} is expired, the LBR forward the buffered MP_i to the interested children (IC). The LBR will only forward the MP_i towards the clusters from which it has not received the MP_i. It is because the interested members in remaining clusters have already received the MP_i during inter-cluster routing. For this purpose, before forwarding the MP_i to its IC, the LBR re-calculate its IC set (refer line 15 in Algorithm 1. It removes the children that belongs to the clusters which have already seen the MP_i in its way up towards the LBR. After re-calculation of IC, if the new IC set is empty, the LBR drops the packet. Additionally, to ensure that the inter-cluster routing will not happen in case where the MP_i is travelling from the LBR to the clusters, we use *C2C* flag bit in IPv6 header of the MP_i. When an intermediate node founds that the *C2C* flag bit is set to 1 in a received packet, then it will perform only the downward routing as the upward routing and inter-cluster routing has already been taken place in past. Please note that in RECOUP all the hop-to-hop data packet transmissions in downward routing are done using an *Optimized Forwarding Mechanism* (OFM) scheme as presented in BMRF protocol (Lorente et al., 2017).

Routing at non-LBR Node. Algorithm 2 shows the working methodology of RECOUP routing protocol when a non-LBR node (say X) has a data packet to send to a multicast destination address. When X sends a multicast packet (say MP_i) as a source node, it goes through the following steps.

- X sets the Pkt_{drop} (i.e., maximum number of forwarding hop-counts for MP_i in inter-cluster routing) equal to the rank of X (X_r). It is done to avoid the routing loops in the network and to control the number of forwards of MP_i, which minimizes the duplicates as well as congestion in the network. The reason that we set Pkt_{drop} to X_r is because after Pkt_{drop} hops the MP_i will reach to the LBR, and then the LBR could simply send the packet to the remaining interested children using downward multicast routing. The Pkt_{drop} is set by the source node only, and it is decreased by each intermediate node until the value reaches to 0 (refer lines 17 to 19 in Algorithm 2).
- X sets the C_{travel}, it consists of a set of CIDs of the clusters in which MP_i has been already forwarded. At the beginning of MP_i's routing (i.e., when it is at the source node), the C_{travel} at node X contains the ID of $X's$ cluster and the ID of the clusters to which the neighbour nodes of X belongs (refer line 4 in Algorithm 2). We do not consider the siblings neighbours because all the siblings have the same cluster ID. If a node has no neighbour than its NC_{ID} set remains empty. The intermediate nodes keep on updating the C_{travel} in MP_i with new cluster IDs before forwarding the MP_i to the nodes with CIDs that are not present in current C_{travel}. Both, the Pkt_{drop} and the C_{travel} values are added in the RPL Packet Information field which is given in IPv6 header format.

Algorithm 2 Non-root node(s) multicast packet routing process in RECOUP.

INPUT at a Node: Data packet
OUTPUT: Forward the data packet towards its destination
1: **if** X has a MP_i to send to MG_i **then**
2: **if** X = Src MP_i **then**
3: $Pkt_{drop} \leftarrow X_r$
4: $C_{travel} \leftarrow X_{CID} \cup NC_{ID}$
5: X forward MP_i towards X_P, X_{IC}, and $X(N_j)$ here 1 to j neighbours of X each with different *CID*
6: **else**
7: **if** X receives MP_i from X_P **then**
8: perform only downward routing
9: **if** C2C = 0 **then**
10: forward MP_i to P, IC, and N_j
11: **else**
12: forward MP_i to IC
13: **end if**
14: **end if**
15: **end if**
16: **if** $X_P \vee X_{IC} \vee X(N_j)$ receives MP_i **then**
17: **if** routing = inter-cluster $\vee$ routing = within $X's$ cluster **then**
18: **while** $Pkt_{drop} \neq 0$ **do**
19: $Pkt_{drop} \leftarrow Pkt_{drop} - 1$
20: Update C_{travel} by adding *CIDs* of N_j that are not present in C_{travel}
21: forward MP_i to P, IC, and N_j
22: **end while**
23: **else**
24: forward MP_i to P and IC
25: **end if**
26: **if** LBR receives MP_i **then**
27: GOTO Algorithm 1
28: **end if**
29: **end if**
30: **end if**

- Once X sets the Pkt_{drop} and C_{travel}, it transmits the MP_i to its preferred parent (P), interested children (IC), and the neighbours (N_j) with different CIDs. In case there are more than one neighbours of X that belongs to a same cluster, X randomly send the MP_i to only one neighbour. This is because the other nodes will receive the MP_i when the intra-cluster[3] routing is performed for MP_i.

In general, to decrease the number of duplicates in the network, a node (including LBR) never forwards a packet to the node or cluster from which the packet has been received. As it can be seen from Algorithm 2 that if the received packet is not a duplicate, the node perform the following steps.

- If the packet is received from a Preferred Parent (refer lines 7 to 12 in Algorithm 2):
  - *Step1:* node checks the packet header for *C2C* flag status, if the flag is not set (i.e., 0), the node is allowed to forward packet to its neighbours that belong to different clusters, else node goes to Step 2.
  - *Step2:* node checks its routing table (or multicast group subscription table) for any interested children that are registered for the multicast address specified in the received packet, and then for-

[3] the intra cluster routing consists of the traditional upward and downward routing techniques that are used for data transmission in RPL's storing mode, i.e., MOP3.

ward the packet to them by using OFM, else the node goes to Step 3.
  - *Step3:* if the node itself is a member of the multicast group given in the received packet, then it sends the packet up to the network stack, else it discards the packet.
- If the packet is received from a neighbour/children (refer lines 16 to 22 in Algorithm 2):
  - *Step4:* node forwards the packet to its preferred parent.
  - *Step5:* node performs the aforementioned steps 1, 2 and 3.
- If a non-root source mote wants to send packet(s), then it will execute the above mentioned steps 1, 2 and 4.

4.2.3. Example of RECOUP routing procedure

For better understanding the routing process of RECOUP, let's consider an IoT network scenario as depicted in Fig. 1. Fig. 1 shows the network state after completion of the DODAG formation and cluster creation phase while executing RECOUP protocol in the *Contiki Cooja* emulator. In Fig. 1, nodes 1, 5, 7, 12, 14, 15, and 16 are the cluster heads of the DODAG, and node 31 is the source node of a multicast group which also include nodes 4, 16, 21, 24, 30, and 38. Assume that the cluster IDs of the clusters is same as the node ID of their clusterhead.

Node 31 executes the multicast packet transmission procedure as follow: (i) it transmits the multicast packet to its preferred parent i.e., 29; (ii) there are no interested children, so it will not send the packet to 32; (iii) in its neighbour set there are only two nodes that does not belong to node 31's cluster, i.e., 34 and 36, but both these nodes belong to the same cluster, so 31 will forward the packet to 34 as it has higher rank than 36. As shown in Fig. 1 that from 29 the packet travel towards the root by executing steps 1–4, meanwhile it also serves all the destination nodes (if any, such as 30) in the way. However, at the same time (i.e., while travelling towards the root), the packet is also disseminated in the various clusters. For instance, upon reception of the packet from node 31, 34 forward it to 6, and it also forward the packet to the neighbour cluster by sending it to 4. Node 34 also send the packet towards 38 through 37 because it is registered with the multicast address given in the packet. In this way, the packet travels vertically as well as horizontally at the same time, thus, it deceases the propagation delay of the packet for their destinations.

It is seen in Fig. 1 that the total number of transmissions (TX_n) required to send a multicast packet from node 31 to all its destinations using RECOUP routing mechanism is 13. While, by using the BMRF and RPL MOP 3, the TX_n required is 17. However, the end-to-end propagation delay is not directly proportional because the packet is travelling in various clusters in parallel. For instance, node 29 forwards the packet to node 30 in parallel when 34 forwards it to 37. The value of TX_n greatly depends on the network topology (i.e., DODAG formation) and the position of source and destination nodes. For example, if we add the node 34 in the multicast group and remove 38, then the TX_n required for RECOUP will decrease by a value of two, i.e., $TX_n = 11$, while for BMRF and RPL MOP 3 it will increase by three, i.e., $TX_n = 20$. We are using TX_n parameter because it affect various other metrics that define the communication robustness and scalability in a network. In particular, lower TX_n implies low end-to-end delay and inherent routing support for scalability.

4.2.4. Optimized Forwarding Mechanism

The RECOUP protocol uses "Optimized Forwarding Mechanism" (OFM) to minimize forwarding of messages during downward routing. In particular, when a parent receives a multicast packet, and it has n number of interested children for the packet, then the parent need to decide whether to send the packet to each children using unicast mode (i.e., create n packets and send one to each child) or to perform a broadcast and all its children will receive the packet. The trade-off between unicast and broadcast mode occurs because the use of unicast mode require more energy consumption as same message is sent by parent for n number of times, while the broadcast makes all the children (including the non-interested ones) to receive the packet in one transmission but it increases communication overhead.

The RECOUP protocol uses OFM as follow:

- Upward forwarding is done using Link Layer unicast because a node has only one preferred parent at any time during communication process. The inter-cluster forwarding is also done by Link Layer unicast as only one node from a neighbouring cluster needs to receive the message to circulate it in the whole cluster.
- Downward forwarding is done based on the Mixed mode decision algorithm proposed in BMRF (Lorente et al., 2017) with three as a threshold value in mixed mode.

Table 2
Simulation parameters for RECOUP protocol evaluation.

Parameters	Values
Emulator	Cooja on Contiki v2.7
Simulation time	10 Minutes
Scenario Dimension	200 × 200 to 800 × 800 sq.meter
Node distribution	Random
Number of nodes	51 to 201 sky motes (including root)
Transport layer protocol	UDP
Number of source motes	8
Routing Protocols	ESMRF, BMRF, and RECOUP
Root waiting timer t	Depends on the value of α
Multicast group or Subscriptions	20%, 40%, 60%, 80%
Radio Medium	Unit Disk Graph Medium (UDGM)
PHY and MAC Layer	IEEE 802.15.4 with CSMA and ContikiMAC
RNG Seed	30 iterations each with new seed
Application protocol	CBR
Transmission Range	25 m
Number of attacker nodes	10%–40%
Traffic rate	0.50 pkt/sec - 500 packets
ESMRF	Contiki v2.7 Default Configuration
BMRF and RECOUP	Mixed mode (Threshold: 3)

5. Simulation and result evaluation

In this section, we present the performance evaluation of RECOUP protocol. We compare it with two state-of-the-art protocols: (i) ESMRF (Fadeel et al., 2015), an enhanced stateless multicast RPL-based forwarding protocol, and (ii) BMRF (Lorente et al., 2017), a bidirectional multicast RPL forwarding protocol. We have fully implemented RECOUP protocol on *Cooja*, the Contiki network emulator (Dunkels), and we used the available open source codes of ESMRF and BMRF for comparison purposes. Table 2 provides the detail of various parameters along with their values that we have used to configure the target LLN scenarios in *Contiki Cooja* emulator.

5.1. Performance analysis

The evaluation metrics used to evaluate the performance of RECOUP are as follow: (i) packet delivery ratio, (ii) end-to-end delay, (iii) per packet energy consumption, and (iv) memory consumption. The values for these metrics are calculated in different network scenarios that are created by varying network size, network traffic, and number of attacker nodes. The source and multicast destination nodes are selected randomly, and the final results plotted are calculated using 30 simulation runs each with different seed value.

5.1.1. Effect of increase in network load

In this section, we discuss how the performance metrics are influenced by the change in the percentage of subscribers (or sinks/destinations) in the target network for the ESMRF, BMRF, and RECOUP routing protocols. It should be noted that in the target network

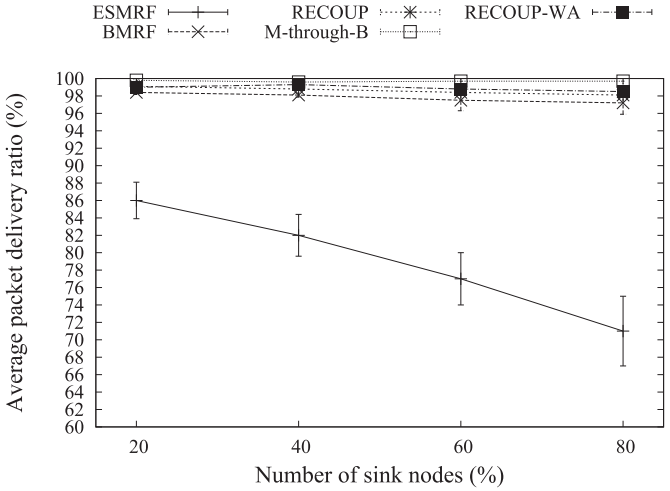

Fig. 2. Packet delivery ratio with increased percentage of sinks.

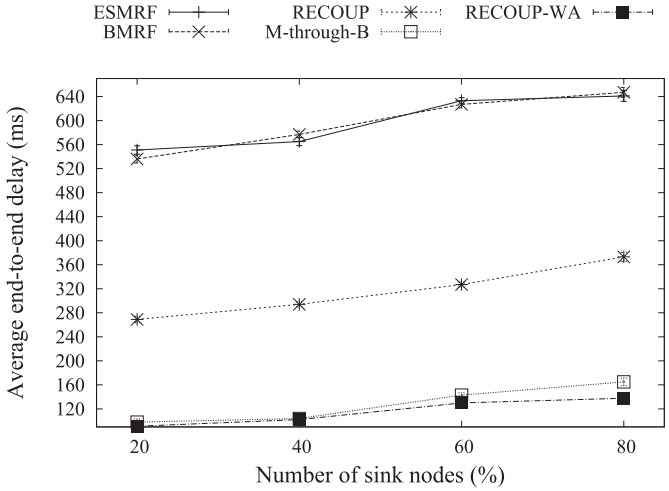

Fig. 3. End-to-end delay with increased percentage of sinks.

scenario we are varying the number of sink node's percentage in the range of 20%–80%, but we keep the fixed number of source nodes (i.e., 8). With the increase in the sink nodes in the network, the network traffic increases as more number of packets are travelling (depending upon the location of the destination) in the network. For these scenarios, we assume that no adversary is present in the network, and the network consists of 100 nodes (excluding the LBR/root). Along with ESMRF and BMRF, we also compare RECOUP's performance against (i) multicasting through broadcasting or simple flooding ($M - through - B$), and (ii) when RECOUP is executed without considering the effect of α (RECOUP-WA), please refer to Equation (1) in which α is used to calculate the packet forwarding waiting time at LBR.

Fig. 2 shows the change in the average Packet Delivery Ratio (PDR) for all the comparing protocols with increase in the percentage of sink nodes. As shown in Fig. 2, the PDR of RECOUP remains higher as compared to ESMRF and BMRF. However, BMRF and RECOUP has nearly the same PDR due to their upward and downward forwarding mechanism, RECOUP has slightly higher PDR due to its inter-cluster forwarding rule. The inter-cluster forwarding helps RECOUP to disseminate the packets even in (small) partitioned network areas. Additionally, RECOUP forward the packets by going around the broken or weak links, which might have been created due to node transmission range or interference. ESMRF has the lowest PDR due to its strict upward and downward forwarding mechanism which increases the number of transmission required to send the packet to all the destinations.

Fig. 3 depicts the effect on average End-to-End Delay (EED) for all the comparing protocols. It can be seen in figure that RECOUP remains too low when compared with the ESMRF and BMRF. It is because for most of the times the packets reaches to its intended destinations without travelling through root. For instance, in Fig. 1, the node 38 receive all its packets in three transmissions, while for ESMRF and BMRF, the packets will have to travel 9 hops before they reaches to node 38. In particular, the use of our efficient inter-cluster forwarding along with the upward and downward forwarding of multicast messages triggers a quick dissemination of packets in the whole network. Due to the aforementioned reason, the energy consumption for per packet (ECP) delivery is also lower in RECOUP as it is shown in Fig. 4. The BMRF has lower energy consumption then ESMRF due to two reasons, first it uses the optimized forwarding scheme, and second it serves the destinations while forwarding a packet towards root node.

Although, the ECP of RECOUP is lower than other comparing protocols, but same is not true when the energy consumption of the whole network is calculated, as it is depicted in Fig. 5. It is because the total number of packet transmissions are higher in RECOUP as a packet might have to travel in a neighbour clusters even in the cases

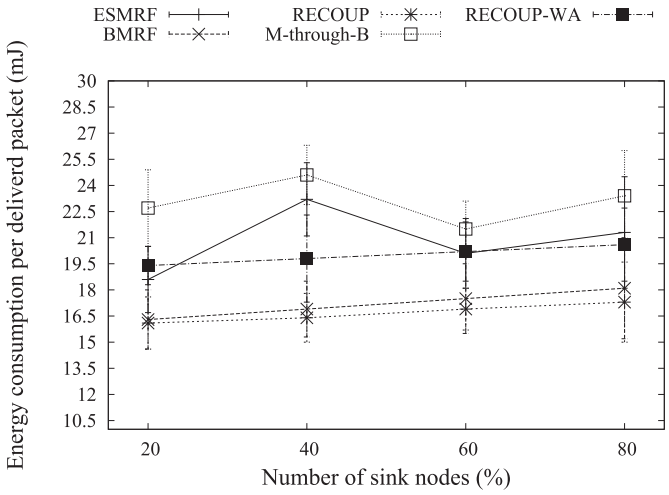

Fig. 4. Energy consumption with increased percentage of sinks.

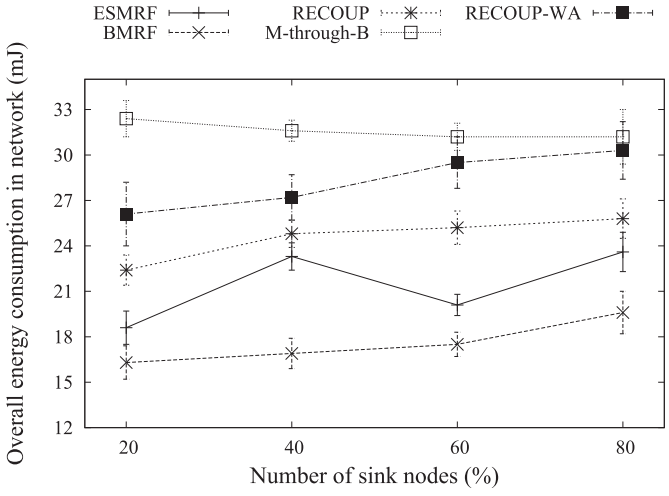

Fig. 5. Overall energy consumption with increased percentage of sinks.

in which no multicast member(s) resides in that cluster. It is the cost that RECOUP pays to achieve improved communication robustness and resistance to an array of routing attacks. However, as the network load increases the additional energy consumption with respect to the net-

Table 3
Effect of RECOUP on some of the derived metrics.

Sink nodes	Hops from S to MD nodes in RECOUP			Hops from Root to MD nodes (without RECOUP)			Cluster with MD nodes
	Min.	Avg.	Max.	Min.	Avg.	Max.	
20	1	2.9	5	1	4	7	6
40	1	3.1	6	1	3.9	9	7
50	1	3	6	1	4.1	8	6
60	1	3.2	7	1	3.6	11	6

work throughput will start decreasing. It is because more and more sinks will start benefiting by RECOUP's inter-cluster routing scheme.

Finally, as depicted in Figs. 2 and 3, both M-through-B and RECOUP-WA achieves equal or more PDR and lower EED than RECOUP. It is because in M-through-B, each node broadcast all the received packets to their neighbours, hence, the packets will eventually be received by their multicast destinations, and the EED will be lower as the packets will travel in all the directions without any restrictions (such as unicast, or inter-cluster). Although, the full broadcast nature of M-through-B will greatly increase the overall network energy consumption as it is shown in Fig. 5. The RECOUP-WA will also have the same PDR as RECOUP but will have lower EED. It is due to lower waiting time (since α is not used to adjust the waiting time) at LBR before it forwards a received data packet to different clusters. However, this will lead to higher number of duplicates in the network as some of the clusters might have already received the packet through inter-cluster routing, and the LBR will resend them those packets. These higher duplicates will increase the energy consumption as it is seen in Fig. 5.

Next, we show the effect of increase in network load on a set of derived metrics in Table 3. These metrics includes, the number of hops between source (*S*) and multicast destinations (*MD*) in RECOUP and without RECOUP, and the number of clusters over which the multicast destinations are spread. The minimum number of hops between S and MD is one as in both the cases (i.e., RECOUP or without RECOUP), one of the multicast destination could be the parent, child, or neighbour. However, the average number of hops to reach all the multicast destination nodes and the maximum number of hops needed to reach the farthest destination for RECOUP are lower. It is because the inter-cluster routing helps to reach to a destination without going through the root node as it is the case when RECOUP is not used. The number of clusters over which the destination nodes are spread will remain same for both the cases. It is because number of sub-DODAGs will be same in case of no RECOUP.

5.1.2. Effect of increased in network size

In this section, we present the effect of increase in network size for all the comparing protocols. Effect of network size is important for the applications where scalability of the network is an important factor. In this scenario, we fix the number of source (i.e., 8) and sinks (i.e., 40%).

The effect on PDR with increase in network size for all the comparing protocols in shown in Fig. 6. It can be seen in Fig. 6 that BMRF and RECOUP protocols remain less affected by the increase in network size when compared to ESMRF. With the increase in network size, the number of transmissions and number of hops between the source and destinations increase greatly, which increases the probability of link break. However, due to inter-cluster routing, the increase in number of intermediate hops in RECOUP is not too high, which helps it to keep the PDR higher even in large networks. The results for the EED in Fig. 7 also support the aforementioned reason as it shows the lower increase in EED for RECOUP with increased network size.

The change in the energy consumption for per packet (ECP) delivery for the comparing protocols with increase in network size is shown in Fig. 8. As the network size increases, the number of duplicate packets also increases for the RECOUP protocol, thus, the ECP increases. It is because, we are also taking into account the energy consumed by

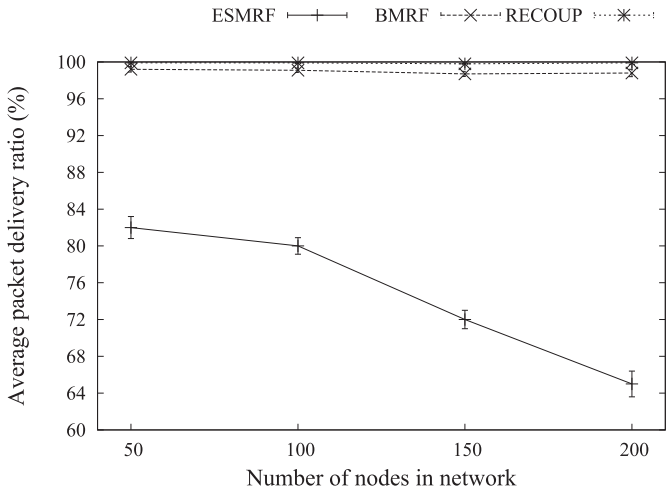

Fig. 6. Packet delivery ratio with increased network size.

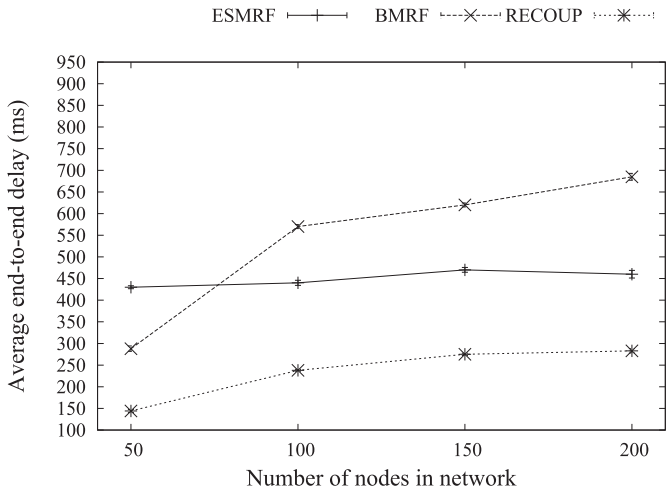

Fig. 7. End-to-end delay with increased network size.

receiving of a duplicate packet at a destination node. The increase in ECP for ESMRF and BMRF is mainly due to increase in the number of transmissions required for a packet to reach to its destinations.

5.1.3. Effect of increase in number of attacker nodes

In this section, we evaluate the performance of RECOUP in presence of attacker nodes in an LLN. We randomly configure nodes to either perform the rank attack followed by selective packet discarding or the blackhole attack. The rank attack could disrupt the correct formation of DODAG, thus, it creates weak or broken links in the network. While, the blackhole attacker will drop all the received packets without forwarding, thus, it decreases the PDR. In LLN, some nodes might select a blackhole node as their preferred parent and start forwarding their data packets through these malicious nodes. The blackhole

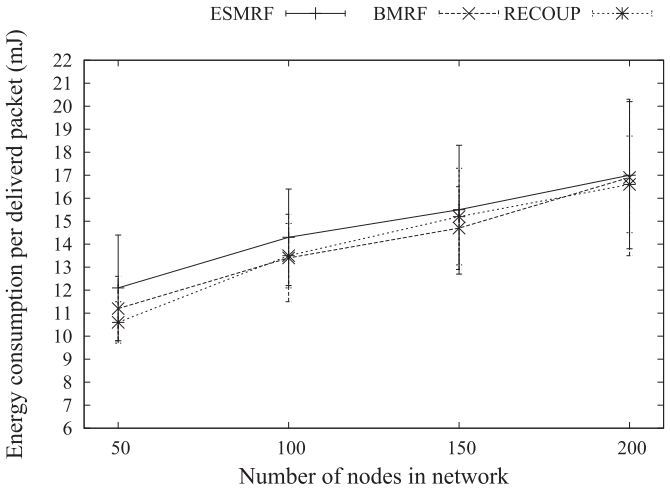

Fig. 8. Energy consumption with increased network size.

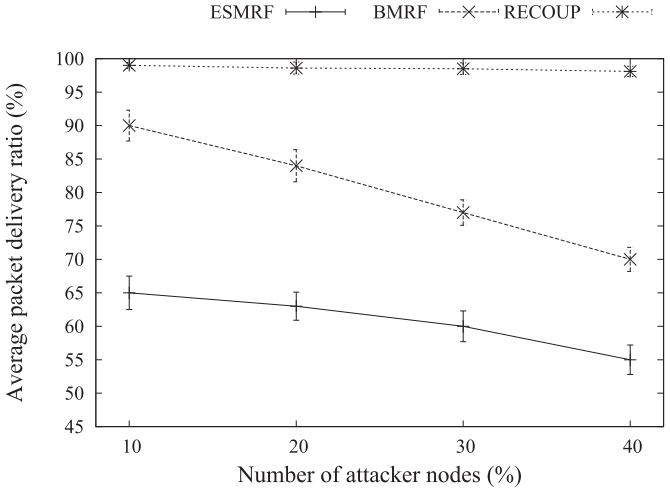

Fig. 9. Packet delivery ratio with increased percentage of attackers.

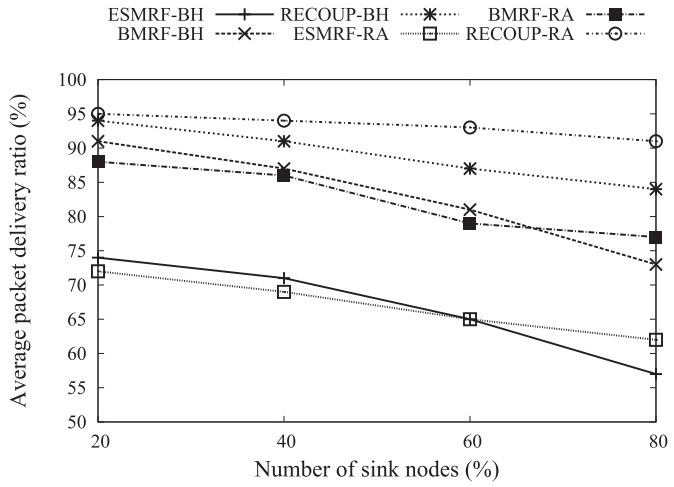

Fig. 10. Packet delivery ratio of comparing protocols in presence of rank and blackhole attacks.

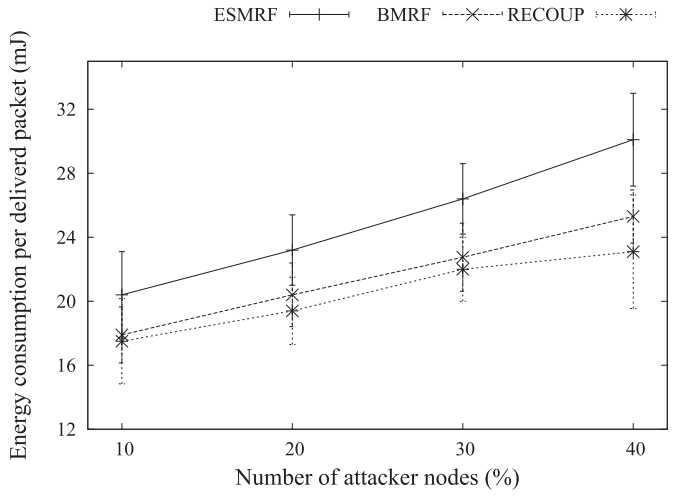

Fig. 11. Energy consumption with increased percentage of attackers.

attack becomes more destructive when it attacks the RPL routing protocol because there exists only one route between a source and destination node in DODAG, and if any one node along the route is malicious, then it could adversely affect the data flow. Moreover, as the rank of the blackhole node decreases (i.e., blackhole node close to LBR/root node) in the DODAG, the chances that it will be part of some active data flow increases.

Fig. 9 show the effect on PDR with increase in number of attackers in LLNs for all the comparing protocols. In this scenario, we set the network size to 101 nodes with 8 source nodes and 40% sink nodes. It can be seen from Fig. 9 that RECOUP has minimal effect of the attacker presence. It is due to its inter-cluster forwarding mechanism that ensures multiple routes to destination nodes. For instance, while using ESMRF and BMRF protocols, if node 29 behaves like a blackhole attacker in the topology given in Fig. 1, then all the packets sent by source node 31 will never reach to its destination nodes. Alternatively, if RECOUP is running as the routing protocol, then all its destinations (except node 30) will receive the packets sent by 31. This behaviour of RECOUP greatly increase its PDR in the presence of attackers. Moreover, since the LLNs exhibit features such as high loss rates, low data rates, and instability, the use of inter-cluster forwarding will improve the data communication in overall network.

The RECOUP only provides communication robustness in the presence of attackers, and it does not detect the attackers. But, the detection can be easily done by using a traffic analysis tool at LBR, which when see that the packets sent by 31 are received through other clusters but not from its own cluster, then it could generate a security alarm. Due to the aforementioned functionality of RECOUP, we can see in Fig. 10 that RECOUP has the highest PDR in presence of both attacks. Also, the position of the attacker greatly affects the PDR as the attacker near the root is much more effective when compared with the attackers residing close to leaf nodes.

Although, we have only tested RECOUP in presence of rank and blackhole attacks, but from the functionality of RECOUP, it is clear that it can effectively minimize the effect of other routing attacks, such as the jamming and sybil attacks. It is because, in RECOUP, the packet travels through multiple routes toward its destination. Thus, the failure or maliciousness of few nodes or links won't effect much to its routing process.

Fig. 11 shows how the increase in number of attackers affect the ECP delivery for all the three comparing protocols. The ECP for ESMRF, BMRF, and RECOUP protocols increases with the increase in attackers. It is because the presence of attackers (mainly the rank attacker) on routing paths increases the number of hop-to-hop re-transmissions. Additionally, the increase in routing path length and disruption in DODAG creation caused by rank attack will leave the network with non-optimal routes. As the RECOUP does not provide any mitigation to these attacks, the increase in the energy consumption per packet shows

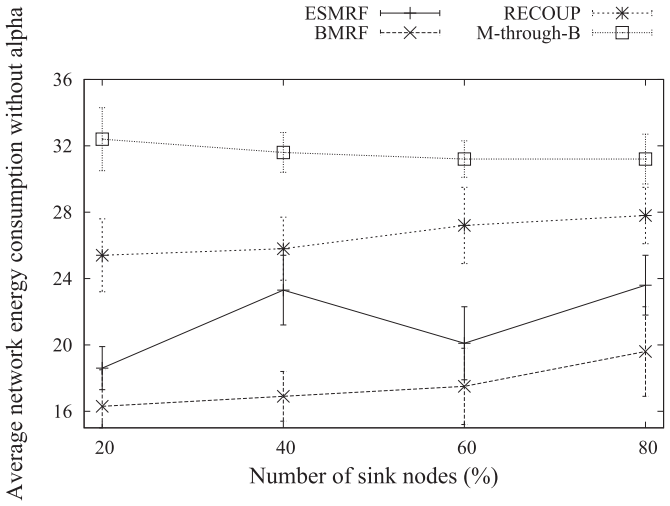

Fig. 12. Overall energy consumption with increased percentage of sink nodes without using parameter Alpha (α).

Table 4
Memory usage.

	Flash [Bytes]	RAM [Bytes]
ContikiRPL	41498	8246
Contiki RPL + Multicast (ESMRF)	948 (+2.3%)	296 (+3.6%)
Improvements for BMRF	250 (+0.60%)	36 (+0.43%)
Improvements for RECOUP	488 (+1.2%)	292 (+3.5%)

the same trend for all the protocols (see Fig. 12).

Finally, Table 4 shows ContikiRPL memory consumption (Corporation), ESMRF, BMRF, and overall code and data memory increase when implementing RECOUP. The memory consumption in RECOUP is slightly higher than the state-of-the-art protocols. To correctly implement all the functionalities of RECOUP protocol, the following additional information is stored at a node: (i) to perform the duplication detection, a node needs to create and maintain the DD_{tab}, which consists of three fields (i.e., message sequence number and source-destination address pairs); and (ii) a new field is added in neighbour table, which consists of entries of its neighbours Cluster ID (C_{id}). The cost of RECOUP is 488 Byte of Flash and 292 Byte of RAM. We consider around 95 to 100 entries for both the tables, which are sufficiently large amount w.r.t a large IoT network. The additional memory consumption in RECOUP compared to the traditional RPL protocol is almost negligible considering the additional features it provides.

5.2. Discussion on data communication robustness and security

The optimized inter-cluster forwarding mechanism used in RECOUP greatly reduces single point of failures in the network, thus, it makes the communication system more robust for data communications. All the existing multicast routing approaches suffer from the scalability issue. It is because as the number of nodes increases in the network, the size of DODAG tree increases which causes increase in the number of hops travelled by a message and it decreases the packet delivery ratio. It further increases the following: (i) probability of a route break, (ii) the energy consumption, and (iii) the end-to-end delay. Apart from the end-to-end delay, the PDR is a critical metric in various application scenarios where sensitive operations are dependent on the information received from other parts of the network. Hence, we believe that providing communication robustness along with the network scalability while keeping in mind the constrained nature of IoT devices is a major challenge for routing protocols in IoT networks, and RECOUP aims to address it.

Communication security is considered as one of the key challenges in IoT networks due to its openness. In Linus Wallgren and Voigt (2013), authors discuss few of the well known security attacks on the RPL protocol which includes selective-forwarding, sinkhole, blackhole, wormhole, clone ID or sybil, and rank attacks (Glissa et al., 2016). Their research shows that the RPL protocol running on top of 6LoWPAN networks is vulnerable to all the aforementioned attacks. Furthermore, all the existing extensions of RPL which includes the ESMRF and BMRF also fails to address or resist any of these security threats in IoT networks. It is because all these protocols use the DODAG topology which contains the possibility of single point of failure. For instance, in Fig. 1, if node 9 is down due to some technical fault or an attack performed by some adversary, all the messages sent by it will never reach to its destination nodes (i.e., 4, 16, 21 and 24). In RECOUP, this will not happen due to the inter-cluster routing which minimizes the impact of above mentioned attacks, and improves the data communication system in the whole network.

In LLN scenarios, depending upon the application requirements, we might have real time deadlines. However, the devices are deployed in an insecure environment, thus ensuring the communication robustness, and on-time and secure communication are crucial aspects. To this end, RECOUP performs the data communication in a way that ensures that it will avoid the single point failures, push the network communication towards scalability, minimize the effect of network partitioning, and reduces the propagation delay for recipients of the data packets. Rather than getting failed in an IoT environment, our protocol works reasonably better and send its data traffic successfully. Due to the inter-cluster communication which triggers a faster dissemination of the information, RECOUP is able to easily mitigate the worst effects of few of the aforementioned attacks. The scalability and quick dissemination features of RECOUP could also be very helpful in enhancing the performance of large scale attestation techniques (Conti et al., 2018) used in IoT networks. These attestation techniques dynamically verifies the integrity of various software and hardware components residing on an IoT device at runtime which increases the network security.

6. Conclusions

In this paper, we propose a new robust multicast routing protocol (RECOUP) for Low-power and Lossy Networks such as 6LoWPAN, which are highly used for deploying IoT networks for various smart service applications. RECOUP uses the advantages of the recently proposed BMRF protocol which includes support for dynamic group registrations and enabling upward and downward forwarding, and it addresses BMRF's key disadvantages such as higher end-to-end delay, low robustness against security attacks (such as rank and blackhole) and link failures, and low scalability. We show that RECOUP remains largely unaffected from rank and blackhole attacks as it delivers more than 80% of data packets to destinations in the presence of attackers. From the simulation results, we can conclude that RECOUP effectively achieve its goals (robustness against security attacks and link failures) at the expense of a slightly higher energy and memory consumption. Please note that RECOUP does not aims to provide explicit security solutions for various attacks (e.g., blackhole, rank attack, selective packet discarding, wormhole, etc), however, it proves that it is robust and effective in data communication process in the presence of these attacks.

As the RECOUP protocol only provide resistance for the security attacks but not the mitigation, in the future work, from the security point of view we are looking to embed algorithms to countermeasure more specific (to IoT network) attacks such as rank and version attacks.

CRediT authorship contribution statement

Mauro Conti: Supervision, Conceptualization. **Pallavi Kaliyar:**

Writing - original draft, Investigation, Methodology, Validation. **Chhagan Lal:** Writing - review & editing, Conceptualization, Supervision.

Declaration of competing interest

The authors declare that there is no conflict of interest of any kind for the submitted manuscript.

Acknowledgements

Pallavi Kaliyar is pursuing her Ph.D. with a fellowship for international students funded by Fondazione Cassa di Risparmio di Padova eRovigo (CARIPARO). This work is also supported in part by EU LOCARD Project under Grant H2020-SU-SEC-2018-832735. The work of M. Conti was supported by the Marie Curie Fellowship through European Commission under Agreement PCIG11-GA-2012-321980.

References

Ahmed, F., Ko, Y.-B., 2016. Mitigation of black hole attacks in routing protocol for low power and lossy networks. Secur. Commun. Network. 9, 5143–5154.

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., Ayyash, M., 2015. Internet of things: a survey on enabling technologies, protocols, and applications. IEEE Commun. Surv. Tutor. 17 (4), 2347–2376.

Conti, M., Kaliyar, P., Rabbani, M.M., Ranise, S., Oct 2018. Split: a secure and scalable rpl routing protocol for internet of things. In: 2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 1–8.

Corporation, M., Ultra low power IEEE 802.15.4 compliant wireless sensor module. http://www.crew-project.eu/sites/default/files/tmote-sky-datasheet.pdf.

Dunkels, A., . [Online]. Available: http://www.contiki-os.org/download.html.

Dvir, A., Holczer, T., Buttyan, L., 2011. VeRA - version number and rank authentication in RPL. In: 2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems, pp. 709–714.

Fadeel, A., Qorany, K., Sayed, E., Khaled, 2015. ESMRF: enhanced stateless multicast rpl forwarding for IPv6-based low-power and lossy networks. In: Proceedings of the 2015 Workshop on IoT Challenges in Mobile and Industrial Systems, ser. IoT-Sys 15, New York, NY, USA, pp. 19–24.

Glissa, G., Rachedi, A., Meddeb, A., Dec 2016. A secure routing protocol based on RPL for internet of things. In: 2016 IEEE Global Communications Conference (GLOBECOM), pp. 1–7.

Hui, J., Vasseur, J., Culler, D., Manral, V., 2012. An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL). [Online]. Available: https://tools.ietf.org/rfc/rfc6554.txt.

Kim, H.S., Ko, J., Culler, D.E., Paek, J., 2017a. Challenging the Pv6 routing protocol for low-power and lossy networks (RPL): a survey. IEEE Commun. Surv. Tutor. 19 (4), 2502–2525.

Kim, H.S., Ko, J., Culler, D.E., Paek, J., 2017b. Challenging the ipv6 routing protocol for low-power and lossy networks (rpl): a survey. IEEE Commun. Surv. Tutor. 19 (4), 2502–2525.

Kushalnagar, N., Montenegro, G., Schumacher, C., 2007. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem State- Ment, and Goals. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4919.txt.

Levis, P., Clausen, T., Hui, J., Gnawali, O., Ko, J., March 2011. The Trickle Algorithm. [Online]. Available: https://tools.ietf.org/rfc/rfc6206.txt.

Linus Wallgren, S.R., Voigt, T., June 2013. Routing attacks and countermeasures in the RPL-based internet of things. Int. J. Distributed Sens. Netw. 11, https://doi.org/10.1155/2013/794326 [Online]. Available:.

Lorente, G.G., Lemmens, B., Carlier, M., Braeken, A., Steenhaut, K., 2017. BMRF: bidirectional multicast RPL forwarding. Ad Hoc Netw. 54, 69–84.

Medjek, F., Tandjaoui, D., Romdhani, I., Djedjig, N., Aug 2017. Performance evaluation of rpl protocol under mobile sybil attacks. In: 2017 IEEE Trustcom/BigDataSE/ICESS, pp. 1049–1055.

Oikonomou, G., Phillips, I., Tryfonas, T., 2013. IPv6 multicast forwarding in RPL-based wireless sensor networks. Wireless Pers. Commun. 73 (3), 1089–1116.

Papadopoulos, G.Z., Georgallides, A., Tryfonas, T., Oikonomou, G., 2017. *BMFA: Bi-directional Multicast Forwarding Algorithm for RPL-Based 6LoWPANs*. Plus 0.5em Minus 0.4emCham. Springer International Publishing, pp. 18–25.

Perrey, H., Landsmann, M., Ugus, O., Whlisch, M., Schmidt, T.C., 2016. TRAIL: topology authentication in RPL. In: Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks, ser. EWSN 16, pp. 59–64.

Pu, C., Hajjar, S., Jan 2018. Mitigating forwarding misbehaviors in rpl-based low power and lossy networks. In: 2018 15th IEEE Annual Consumer Communications Networking Conference. CCNC, pp. 1–6.

Raoof, A., Matrawy, A., Lung, C., 2019. Routing attacks and mitigation methods for rpl-based internet of things. Secondquarter IEEE Commun. Surv. Tutor. 21 (2), 1582–1606.

Romdhani, I., Al-Dubai, A., Qasem, M., Thomson, C., Ghaleb, B., Wadhaj, I., 2016. Cooja Simulator Manual. Tech. Rep.. [Online]. Available: http://researchrepository.napier.ac.uk/Output/299955.

Shelby, E., Zach, Chakrabarti, S., Nordmark, E., November 2010. Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs). [Online]. Available: https://tools.ietf.org/rfc/rfc6775.txt.

Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, J., Alexander, R., 2012. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6550.txt.

Zarpelo, B.B., Miani, R.S., Kawakani, C.T., de Alvarenga, S.C., Apr. 2017. A survey of intrusion detection in internet of things. J. Netw. Comput. Appl. 84, 25–37 no. C.

Mauro Conti is Full Professor at the University of Padua, Italy, and Affiliate Professor at the University of Washington, Seattle, USA. He obtained his Ph.D. from Sapienza University of Rome, Italy, in 2009. After his Ph.D., he was a Post-Doc Researcher at Vrije Universiteit Amsterdam, The Netherlands. In 2011 he joined as Assistant Professor the University of Padua, where he became Associate Professor in 2015, and Full Professor in 2018. He has been Visiting Researcher at GMU, UCLA, UCI, TU Darmstadt, UF, and FIU. He has been awarded with a Marie Curie Fellowship (2012) by the European Commission, and with a Fellowship by the German DAAD (2013). His research is also funded by companies, including Cisco, Intel, and Huawei. His main research interest is in the area of Security and Privacy. In this area, he published more than 300 papers in topmost international peer-reviewed journals and conference. He is Area Editor-in-Chief for IEEE Communications Surveys & Tutorials, and Associate Editor for several journals, including IEEE Communications Surveys & Tutorials, IEEE Transactions on Information Forensics and Security, IEEE Transactions on Dependable and Secure Computing, and IEEE Transactions on Network and Service Management. He was Program Chair for TRUST 2015, ICISS 2016, WiSec 2017, ACNS 2020, and General Chair for SecureComm 2012 and ACM SACMAT 2013. He is Senior Member of the IEEE.

Pallavi Kaliyar is currently a Ph.D. student in school of Brain Mind and Computer Science at the University of Padova, Italy with a fellowship for international students funded by Fondazione Cassa di Risparmio di Padova e Rovigo (CARIPARO). Here, she is part of the SPRITZ Security and Privacy Research Group research group under the supervision of Prof. Mauro Conti. She received my Masters of Technology in Computer Science and Engineering in 2012 and Bachelor of Engineering in Computer Science and Engineering in 2008. She is conducting research on fields including security and communication reliability related to the Internet of Things and Software Defined Networking.

Chhagan Lal is currently working as a postdoctoral research fellow at Simula Research Laboratory, Norway. Previously, he was a postdoctoral fellow at Department of Mathematics, University of Padua, Italy, where he was part of the SPRITZ research group. He received his PhD in Computer Science and Engineering from the Malaviya National Institute of Technology, Jaipur, India, in 2014. During his PhD, he has been awarded with the Canadian Commonwealth Scholarship under the Canadian Commonwealth Scholarship Program to work in University of Saskatchewan, Saskatoon, SK, Canada. His current research areas include applications of blockchain technologies, security in softwaredefined networking, and Internet of Things networks.