
Article:
Chu, Yi, Mitchell, Paul Daniel orcid.org/0000-0003-0714-2581, Grace, David orcid.org/0000-0003-4493-7498 et al. (3 more authors) (2023) IRIS: A low duty cycle cross-layer protocol for long-range wireless sensor networks with low power budget. Computer networks-The international journal of computer and telecommunications networking. 109666. ISSN 1389-1286

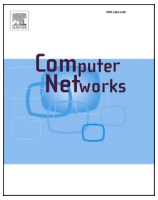
https://doi.org/10.1016/j.comnet.2023.109666

Computer Networks 225 (2023) 109666

Contents lists available at ScienceDirect

Computer Networks

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



IRIS: A low duty cycle cross-layer protocol for long-range wireless sensor networks with low power budget

Yi Chu [a,*], Paul Mitchell [a], David Grace [a], Jonathan Roberts [b], Dominic White [b], Tautvydas Mickus [a]

[a] School of Physics, Engineering and Technology, University of York, York, United Kingdom
[b] TE Connectivity UK Ltd, Swindon, United Kingdom

ARTICLE INFO

Keywords:
Wireless Sensor Network
cross-layer protocol
energy constraint
duty cycle

ABSTRACT

This paper presents a lightweight cross-layer protocol relIable Routing with coordInated medium acceSs control (IRIS) which is designed for long-range pipeline Wireless Sensor Networks (WSNs) with extremely low power budget, typically seen in a range of monitoring applications. IRIS includes functions of network discovery, medium access control and routing designed for heavily energy constrained networks. The protocol is able to operate with less than 1% duty cycle, thereby conforming to ISM band spectrum regulations in the 868MHz band. The duty cycle can be flexibly configured to meet other regulations/power budgets as well as to improve the route forming performance. Simulation results show a guaranteed route formation in different network topologies with various protocol configurations. An analytical model is presented to validate the simulation results of route formation time. System robustness against unreliable wireless connections and node failures are also demonstrated by simulations. A network of 71 attenuated LoRa nodes is implemented to evaluate the capability of IRIS. Long-term operation of the network shows consistent performance compare with the simulations.

1. Introduction

Wireless Sensor Networks (WSNs) offer low-cost solutions for long-term monitoring tasks in many different environments [1]. Compared with conventional wired monitoring systems, wireless sensor nodes can be rapidly deployed with minimal infrastructure requirements, with specifically designed protocols the WSNs have the potential to automatically form a network according to their self-organising nature. During this operation, WSNs can be designed to be robust to single node failures, thereby being exempt from frequent maintenance by human engineers. In recent decades, WSNs have been applied to many applications, such as geological event monitoring [2], animal habitat monitoring [3,4], oil and gas industry [5], health monitoring [6], smart cities [7], smart grids [8], and smart farming [9]. Now WSNs have become the solid foundation of the rapidly expanding Internet of Things (IoT) [10].

Wireless sensor nodes are normally powered by batteries or energy harvesting devices [11], which makes energy efficiency a critical requirement where WSNs need to operate for a long period of time. The Radio Frequency (RF) module on a node is usually a major energy consumer, and the power consumption is mainly from (re)transmitting, receiving packets, and idle listening. Energy consumed by packet collisions, retransmissions, exchanging control information, idle listening, and overhearing are considered overheads [12]. Well-designed Medium Access Control (MAC) and network layers should be able to keep these overheads as low as possible, while achieving the performance required by the application. For example, the well-established Sensor-MAC (S-MAC) [13] and Zebra MAC (Z-MAC) [14] use duty cycling procedures to switch nodes between active and sleep states to conserve energy. Some regulations have also considered limiting the duty cycles for spectrum sharing purposes. For example, the UK regulator Ofcom's IR 2030 document [15] has limited the duty cycle of most 868 MHz industrial, scientific and medical (ISM) bands to 1%, where such a band is widely used by Long-Range Wide-Area Network (LoRaWAN) [16] nodes for many IoT applications [17].

Having a low duty cycle benefits energy efficiency while meeting the regulations, however it limits the number of packet exchanges between nodes, which are necessary for operating the MAC and network layer protocols. Motivated by these constraints, in this paper we present a

The research was funded and supported by TE Connectivity.
* Corresponding author.
E-mail address: yi.chu@york.ac.uk (Y. Chu).

https://doi.org/10.1016/j.comnet.2023.109666
Received 11 May 2022; Received in revised form 31 January 2023; Accepted 27 February 2023
Available online 3 March 2023

simple but novel cross-layer protocol named relIable Routing with coordInated medium acceSs control (IRIS) which is designed to operate with an ultra-low duty cycle, for long-range monitoring tasks such as river/canal monitoring, coastline monitoring, underwater cable, motorway and railway monitoring, oil pipeline monitoring and power transmission line monitoring [18]. The conventional methods for such tasks normally involve human intervention and the general timescale for collecting one round of monitoring data could be weekly, monthly, or even longer [19]. The WSN approach can significantly reduce timescales and provide much more frequent observations. These monitoring applications require pipeline network topologies (potentially over hundreds of kilometres), often in environments that are not well served by existing wireless infrastructure, so the monitoring information must be passed over a large number of hops to reach the destination. The organisation of routes through which information travels becomes challenging in such situations, where pipelines are long and involve a large number of nodes. The proposed IRIS protocol achieves neighbourhood discovery and integrates energy efficient MAC and network layers, achieved by simple node logic to achieve operation with a less than 1% duty cycle. The 1% duty cycle is set as a target to both meet the RF regulations and keep the energy consumption as low as possible. Specific contributions of this paper are summarised as follows:

1) A protocol which can operate in a scenario where the duty cycles of all nodes are less than 1% during both network initialisation and normal operations, which allows the network to operate under ultra-low energy budget. Unlike many state-of-the-art protocols which require the nodes to have high duty cycles during initialisation, IRIS can be deployed to nodes with extremely limited initial energy storage (e.g., nodes powered by energy harvesting devices only without batteries). As far as we are aware, no other routing or cross-layer protocols guarantee route finding with such low duty cycle during network initialisation. This is also the reason why there are no simulations results of route formation time from other protocols present in Section 5 which evaluates the performance.
2) A protocol where the duty cycle can be controlled. IRIS uses ping packets periodically initiated by a base station at one end of the network (and relayed by multiple nodes on a route) to carry monitoring information to a base station at the other end of the network. This allows the base station to control the duty cycle of the network according to different requirements. Ping packets are widely used in networks using the Internet Protocol (IP) to test the reachability to certain destinations. In the IRIS protocol, the ping packets will ultimately reach the end base station of the WSN. The functionality of ping packets is significantly extended beyond that used in conventional networks to carry out the tasks of route finding, monitoring data forwarding and loose synchronisation.
3) An energy efficient route discovery process. Route finding is completed while relaying the ping packets without any knowledge of neighbour nodes. The geolocations and knowledge of the duty cycle of neighbour nodes are not required during any operation stage. The simple nature of IRIS allows the protocol to be scaled to networks with different numbers of nodes and hops. The low computational requirements make the protocol feasible for deployment on low-cost nodes.
4) A protocol which integrates coordinated MAC layer and network layer functions under energy and bandwidth constraints. The MAC layer strictly controls the capability of the nodes to initiate packet transmissions thereby ensuring the network layer to operate without suffering from packet collisions. The node logic is able to achieve energy efficient transmission/reception scheduling while relaying the monitoring information towards the base station and to adapt to network topology changes. The simple nature of the protocol allows the network to operate with extremely low available date rates.

The rest of the paper is organised as follows. Section 2 presents the related work on low duty cycle protocols. Section 3 introduces the network topology and the main idea of IRIS. Section 4 explains the methods IRIS uses to achieve reliability. Section 5 presents the performance results and discussions. Section 6 describes the implementation details and observed performance. Section 7 concludes the paper.

2. Related work

Duty cycling is widely used by many MAC protocols to switch wireless sensor nodes into a lower power sleep state whenever possible to conserve energy. For example, sensor-MAC (S-MAC) [13] sets up synchronised sleep schedules across a neighbourhood to make sure nodes wake up at the same time to exchange packets. During active periods, nodes use Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) with Request to Send (RTS)/Clear to Send (CTS) handshakes to contend for transmissions. Many variants [20–24] of S-MAC have been developed to solve the restriction of a fixed duty cycle by considering different service requirements to improve the energy efficiency, but reaching ultra-low (e.g., 1% or lower) duty cycle operation remains difficult.

Another MAC protocol with synchronised duty cycle Scheduled Channel Polling (SCP) [25] has achieved an operating duty cycle of less than 1% in optimal conditions by significantly reducing the length of preambles used in Low-Power Listening (LPL). However, SCP-MAC requires a considerable amount of energy during initialisation for the purpose of synchronisation. X-MAC [26] has tried to tackle the same problem of LPL by using multiple short preambles and acknowledgements (ACKs) from the receiving node to initiate transmissions earlier without synchronised duty cycles. Duty cycles of 5% to 10% are achieved by X-MAC and higher overheads are incurred to compensate for the asynchrony. Another asynchronous protocol AP-MAC [27], uses randomly transmitted beacons to broadcast node sleep schedules to neighbours, so that the sender can predict the receiver's next active period to reduce overheads. However, AP-MAC also requires a significant amount of initial energy, and its lowest duty cycle is approximately 10%.

MAC and network layers working collaboratively benefits the overall performance of a WSN. Many cross-layer protocols have been developed to exploit the information from both layers to achieve better performance. For example, Routing enhanced MAC (RMAC) [28] exploits the routing information between source and destination nodes to effectively switch the relay nodes between active and sleep states to save energy. RMAC also utilises the broadcast nature of wireless signals by sending a single Pioneer Control Frame (PION) message to replace the ACK to the previous hop and the RTS to the next hop. RMAC has achieved a 2% to 3% duty cycle during normal operation but the route finding is not included. Light-Weight Opportunistic Forwarding (LWOF) [29] uses the location information of neighbour nodes to efficiently forward packets to the sink. However, the duty cycle is relatively high (5% to 18%) because of the preambles of LPL and Global Positioning System (GPS) modules, both of which are not always available for WSNs. Dynamic Switching-Based Reliable Flooding (DSRF) [30] has developed the Automatic-Repeat-Request (ARQ)-based flooding tree by using the topology information of parent, sibling and child nodes to align their sleep schedules to reduce overhearing, thereby improving the energy efficiency. DSRF has achieved a 1% duty cycle with the assumption of knowing the duty cycles of all neighbour nodes. The authors of [30] have later proposed a distributed Minimum-Delay Energy-efficient flooding Tree (MDET) [31] network layer algorithm to construct an energy optimal flooding tree. However, energy efficiency is not considered during route finding and the nodes need to be active when constructing the flooding tree. The work in [32] has summarised some state-of-the-art cross-layer protocols that aim at prolonging network lifetime. Similarly, it is rare for the existing work to consider extremely limited initial energy and the solutions to mitigate the issue. Some work in [32] also utilise the nodes' location information to reduce energy

consumption during routing.

Although some state-of-the-art protocols are able to achieve low duty cycle while operating in steady state, the consistently low energy cost during all phases of operations is yet to be addressed. High energy consumption during network initialisation and route finding will significantly reduce the applicability of the protocols to networks without sufficient initial energy storage (e.g., networks powered by energy harvesting devices). The IRIS protocol tackles the problem by maintaining the ability to operate under a low energy budget across the lifespan of the network. This unique feature initiates the possibility of further reducing the cost of the WSNs (batteries are not required) while maintaining long lifetime.

3. Iris protocol design

Long-term and long-range monitoring tasks are always challenging for WSNs because of the constraints of energy, processing power and connectivity. Motivated by these constraints, IRIS uses a simple but novel approach to complete these tasks with extremely low energy availability. This section describes the network topology of the applications that IRIS is designed for, and how IRIS integrates the MAC and network layers.

3.1. Network topology

The target applications of IRIS have a pipeline type of topology such as rivers, coastlines, underwater cables, motorways, and railways. For the purpose of generality, the low-cost nodes are assumed to be randomly deployed along the pipeline and the monitoring information from all source nodes is forwarded to a base station (end base station, represented by control centre in Fig. 1) at the end of the network over a large number of hops. A second base station (start base station) is deployed at the other end of the pipeline for the purpose of initiating communication along the pipeline. The base stations are located far away from each other, but they have direct links to Internet gateways so that the monitoring information can be sent to a remote control centre. In practice, with the exception of a more substantial power source, the base stations could employ the same technology as the low-cost nodes along the pipeline, because the required processing capability of the base stations is not significantly different from other nodes in the network. Infrastructure (e.g., power, network access) is only required at the base stations and the rest of the network is able to operate independently. The IRIS protocol is designed to allow the network to operate without any prior neighbourhood information and without any predetermined structure. The low-cost nodes have no knowledge of their geolocations. The nodes can be deployed in any physical order at random locations. The only fundamental requirement is the need for nodes to be deployed within reasonable radio range such that end-to-end connectivity is achievable. Fig. 1 shows an example of the proposed WSN.

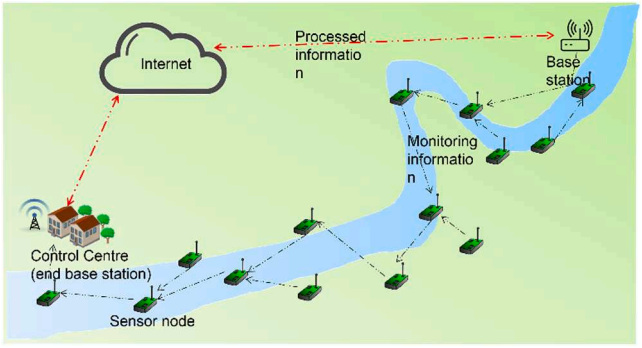


Fig. 1. Network topology.

3.2. MAC layer design of the IRIS protocol

IRIS uses ping packets initiated by the start base station to propagate through the pipeline network while collecting monitoring information from the nodes who can hear the pings. The ping packets ultimately reach an end base station (at the other end of the network) which is connected to the start base station via an external network. Each pipeline network consists of two base stations (start and end base stations) and only the start base station emits ping packets. A configurable active / sleep duty cycle is used to allow the nodes to remain in the low-power sleep mode as long as possible to conserve energy. The active state is further divided to control the time that a node switches between the transmission mode and the reception mode to ensure packet delivery. This allows the IRIS to be applicable to low-cost nodes with only half-duplex RF modules. We refer to the time period of one duty cycle as a frame and the frame length is the same across all nodes in the network. One frame is further divided into a number of slots, where the length of a slot allows a node to send a ping packet and receive an acknowledgement (ACK) packet. The slot length varies according to the different requirements, and the frame size can be adjusted based on the power availability and the duty cycle requirements.

Fig. 2 shows an illustrative example of the MAC layer of the IRIS protocol. This example shows a 6-node section in the middle of a multi-hop WSN. Nodes A, D and F, which relay the ping packets, are defined as *route nodes* (marked in red), and nodes B, C and E, which report monitoring information to the route nodes are defined as *non-route nodes* (marked in blue). We assume all packets are broadcast on the same frequency, so hearing multiple packets simultaneously will cause collisions. Before the example starts, we assume that node A has already received a ping packet from its previous-hop route node and there is already a route formed within this network subsection. More details of the route formation will be explained in subsection C. In slot 1 node A sends a ping packet (marked in red) to node D and receives an ACK packet (marked in green). Node A keeps listening in slot 2 in case other nodes want to report their monitoring information after hearing the ping packet (node B could report in this example). Node A switches to a low power sleep state after the listening period. Nodes B and D adjust their active period in the next frame to start slightly earlier than slot 1 after receiving the ping packet from node A to loosely synchronise to node A and to compensate any potential clock drift. Node D can switch to the sleep state during slot 2 however this may be inefficient for some hardware designs [33] because node D will be active again in slot 3. In slot 3, node D sends the ping packet to node F and receives an ACK. Node E hears the ping packet and has some information to report, so it sends a report packet (marked in blue) to node D in slot 4 and receives an ACK. Node D switches to the sleep state after slot 4. Nodes C, E and F perform the same loose synchronisation after receiving a ping packet from node D. A route node wakes up in the same slot every frame and expects a ping packet from the previous hop. In practice a route node can wake up slightly earlier and adjust its local timer according to the received ping

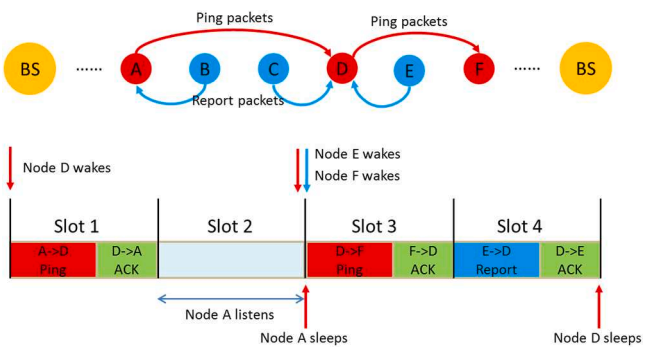


Fig. 2. IRIS MAC layer example.

packet to compensate for potential clock drift.

The node which has not heard a ping packet is defined as a searching node. A node normally starts as a searching node after powering on or losing connection to an existing network. For a searching node to join an existing network, it starts by randomly selecting a slot to listen to. The number of successive slots it continues listening in depends on the energy available. If it does not hear any ping packets during the active period in the current frame, it shifts its active period forward in the next frame. For example, if a node listens to slots 1 to 4 in the first frame and does not hear any ping packets, it listens to slots 5 to 8 in the next frame. Once it hears a ping packet, it can either latch on to the sender of the ping in every frame or keep shifting the active period forward until it cannot hear a ping packet then loop the active period within the slots that it can hear ping packets. For example, in Fig. 3 the searching node is active in slots 9-12 in frame 1 and hears a ping packet in slot 11 then becomes a non-route node (assuming there is already an established route). In frame 2 it can either stay active in the same slots 9-12 (option 1) or shift the active period to slots 13-16. In the former case the node only reports to one route node and in the latter case the node can potentially report to multiple route nodes which could benefit the contention but with a slight risk of losing synchronisation with route nodes with the presence of large clock drift. In the rest of the paper, we assume the nodes follow the activity in option 1, however, selecting either option have minimal effect to the route formation.

3.3. Network layer design of the IRIS protocol

The only information required by IRIS to find a route between two base stations is a unique node ID (e.g., unique integers) configured before deployment and there are no specific requirements during the deployment (i.e., it is not necessary for the nodes to be deployed in any particular order and location information is not required). The only information the nodes require is the pre-programmed slot duration, the number of slots per frame and the number of slots they should remain active for, all which stem from the system design for a particular deployment to meet regularity and energy budget requirements. Energy consuming neighbour discovery is not needed by IRIS. The route formation starts with one base station at one end of the network broadcasting a ping packet. The timing of this ping packet transmission defines the timing (and start) of a frame. Fig. 4 shows the structure of the ping packet, which includes a 4-bit integer packet type, 11-bit integer IDs of the sender and destination (next-hop) nodes, a 1-bit binary link ID which indicates whether the route is formed, and the payload reserved for monitoring information. If the sender does not have an intended destination node, it just includes "-1" in the field of destination ID indicating that it is looking for a next hop to join the network.

While the base station is broadcasting ping packets in the first slot of every frame, other nodes wakeup at random times and listen for the defined active period (a specific number of slot durations) and then go back to sleep for the defined sleep period. If a node does not hear anything during its active period, it delays its active period such that it wakes up a certain number of slots later in the next frame. This process allows all nodes to search for a travelling ping packet without any knowledge about the network topology or timing. The sliding active period also guarantees that the searching node can always find a ping

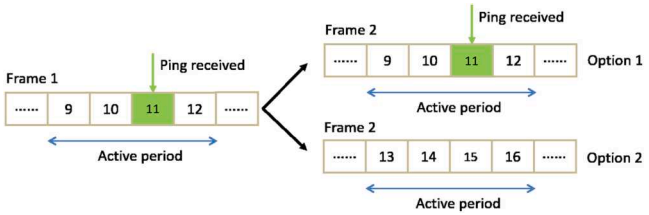


Fig. 3. Example of a non-route node shifting its active period.

Fig. 4. Ping packet structure.

packet, if there is one within its reception range. Once a node hears a ping packet with "-1" destination ID (the sender is defined as a route-end node), it replies with an ACK (including its own ID) to declare its interest in joining the route to relay ping packets. In the next frame, it configures its active period to start in tandem with the timing of the ping packet it received and expects another ping. If the destination ID is still "-1", it configures its active period to start with a random slot in the next frame to avoid potential ACK collisions. If the destination ID is not "-1" nor its own ID, it becomes a non-route node and follows the activities as described in Fig. 3. If the destination ID of the ping packet now becomes its own ID, it means the sender has selected it as the next hop and it becomes the new route-end node. Then the new route-end node repeats the behaviour of its previous hop node by sending ping packets with a "-1" destination ID and expecting other nodes to join the route. This process repeats hop-by-hop until the base station at the other end of the network receives a ping (we assume the network has one start base station and one end base station). The end base station can then inform the start base station of the reception of the ping so both base stations know that a route has been formed and the next ping packet initiated by the base station will have "1" in the link ID field indicating that the route has been formed. This also indicates that the non-route nodes (e.g., nodes B, C and E in Fig. 2) can start to report their monitoring information periodically to the route node after hearing a ping. The basic operating states of the base station which originates the ping packets are shown in Fig. 5. The basic operating states of all other nodes are shown in Fig. 6. Note that only the base station can change the Link ID field in the ping packets.

4. Iris protocol reliability methods

The node logic described in Section 3 establishes the basic MAC and network layer operations. However additional features are needed to secure the reliability requirements. This section demonstrates the methods IRIS uses to avoid routing loops, to make sure the route propagates forward, and to achieve robustness against unreliable wireless connections.

4.1. Inefficient routing avoidance

First of all, one node is able to relay at most one ping packet per frame given its limited active period (forced by the low duty cycle operation), so creating a loop via the same node is naturally impossible for IRIS. However, without information about neighbour nodes and their locations, it is possible for the route to geographically loop or even propagate backwards towards the base station which initiates the ping packets. For example, in Fig. 7 the route geographically propagates backwards at node D and E.

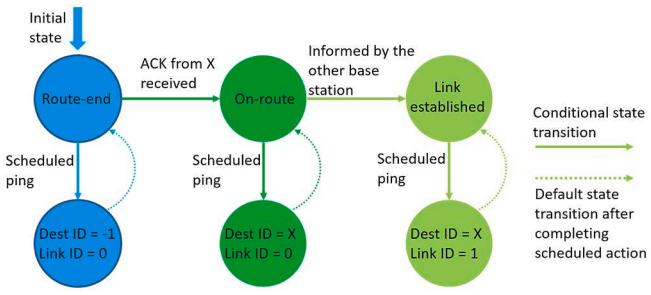


Fig. 5. Base station operating states

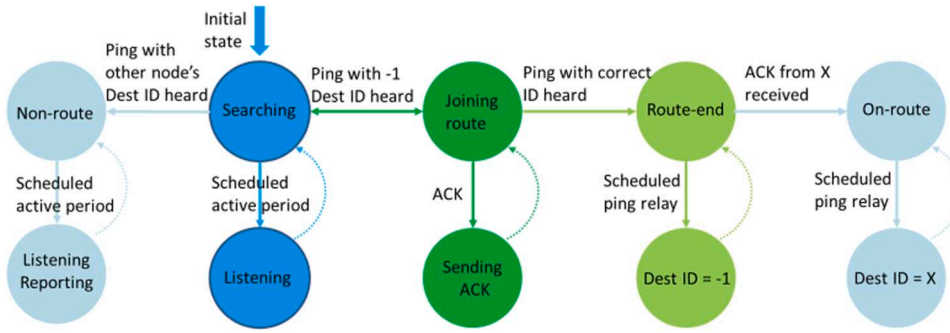


Fig. 6. Node operating states.

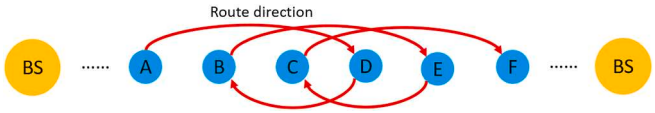


Fig. 7. Some hops propagate backwards.

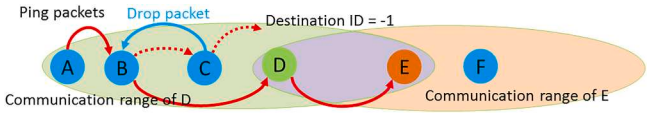


Fig. 9. Example of the drop packet.

To avoid such inefficient routes, the density of the route nodes can be controlled by preventing a node from joining the route when it hears a lot of activity in the neighbourhood. When a node in a searching state shifts its active period forward in each frame, it counts the number of ping packets with non "-1" destination ID it has heard. If this number exceeds a certain threshold, it will not attempt to join the network. We define this threshold as connection limit (*conlimit*), and it is configurable. For example, in Fig. 8, node B is able to hear the ping packets addressed to nodes C and D, and also the ping packet with –1 destination ID from node D. If the *conlimit* is set to 2, node B will not attempt to join the route, thereby avoiding the potential of creating an inefficient route. This *conlimit* also helps finding the next hop relatively far away geographically, thereby reducing the total number of hops needed to reach the other base station.

In the situation where a route-end node is not able to find a next hop to propagate the ping packets (caused by either some neighbour nodes being limited by *conlimit* or because a node does not have a next hop node within radio range), the node sends a drop packet (which has the similar format to an ACK but with a different packet type) to its previous hop and switches to the searching state. Then the previous hop node becomes the route-end node which broadcasts ping packets with a "–1" destination ID to try and find an alternative next hop. We set a threshold *frameout* for the nodes to decide when to send the drop packet. The threshold *frameout* is defined by the number of consecutives frames that the route-end node cannot find a next hop node. For example, in Fig. 9 when node C is the route-end node, nodes D and E can hear two ping packets with non-"-1" destination when they are active thereby will not attempt to become route nodes. When *frameout* of node C exceeds a certain value, it sends a drop packet to node B to find an alternative route.

4.2. Robustness against unreliable wireless connections

In a practical environment, the wireless connections can be subject to interference from other devices using the same spectrum, or the channel could be in a deep fade. Either of these conditions will cause the packets to be partially received/damaged or completely lost. Moreover, the low-cost nodes can stop functioning due to a flat battery or physical damage, which can also cause packet losses. The actions that IRIS will take according to the losses of different types of packets are as follows.

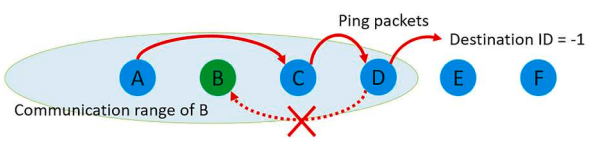


Fig. 8. Example of conlimit = 2.

1) ***Ping packet loss***: a node does not hear a correct ping (with destination ID) when it expects one. If the node is a route-end node, missing a ping packet could indicate either the previous hop is not functioning, or it is simply experiencing a bad connection. The route-end node will generate a ping packet and send it to the next hop as if it heard the ping packet from the previous hop, so that the downstream route is still kept alive in case of minor ping packet losses. At the same time, it increases a counter called *phq_frameout* (previous hop quiet) which indicates the number of consecutive frames that the route node misses ping packets from the previous route node. Once the counter exceeds a certain threshold, the route (-end) node stops generating ping packets and switches back to the searching state. If the node is a non-route node which has latched onto a route node to send its report packets, it will increase a counter called *rq_frameout* (route quiet) in a similar fashion and switch to a searching state once the counter expires.
2) ***ACK (which replies to a ping with destination ID) packet loss***: a node does not hear an ACK from the next hop after it sends a ping. This indicates that either the next hop is not functioning or there is a bad connection. The route node increases a counter called *nhq_frameout* (next hop quiet) and switches to the route-end state once the counter expires and starts sending ping packets with "-1" destination ID.
3) ***ACK (which replies to a ping with "-1" destination ID) packet loss or collision***: an ACK is lost or collided when a searching node attempts to join the route. This does not affect the route-end node which originates the ping packet. If the searching node which sends the ACK wakes up in the same slot next frame and hears the ping with "-1" destination ID again, it configures its active period to start with a random slot in the next frame.
4) ***Report packet and its associated ACK losses***: a non-route node may experience collision or channel fading when sending the report packet to the route node, or the route node sending the ACK back. These do not affect the sleep schedule of the non-route node. Its active period remains as long as it can hear the ping packets.
5) ***Drop packet and its associated ACK losses***. The route-end node will keep sending the drop packets in the same slot every frame until it is

successfully acknowledged if it can hear the ping from the previous hop.

With all the aforementioned features, the node operating states shown in Fig. 6 are updated, as shown in Fig. 10. The node (especially route node) failures can be considered as "permanent" packet losses. The failure of a route node will cause the downstream route to be reconstructed by following the operating states in Fig. 10 while the upstream route remains functioning. A patent has been filed to cover the above features of IRIS [34].

5. Performance and discussions

In this section we evaluate the performance of IRIS with Monte Carlo simulations using Matlab. An analytical model is presented to estimate the route formation time to match the simulation results. The network layer metrics to be evaluated include the time required to establish a route and to recover from node failures, and the robustness against clock drift. The MAC layer metrics to be evaluated include the throughput and delay for the monitoring information to reach the base station after the route is established, and the robustness to unreliable wireless connections.

5.1. Simulation parameters

We assume the nodes are equipped with the LoRa transceiver Semtech SX1276 [35] and the Texas Instruments MSP430F2617 microcontroller [36]. The LoRa transceiver operates in the suggested low-power, low-bitrate and long-range mode. The properties of the LoRa transceiver are listed in Table 1. The microcontroller has a current draw of $I_{act} = 5.84$ mA when active at 16 MHz (could be lower for slower processor speed) and $I_{std} = 0.5$ μA in standby mode. The nodes are also equipped with a super capacitor charged by an energy harvester to power the transceiver and the microcontroller. We consider that a centimetre-level size wind energy harvester is used such as the one presented in [37] which is able to produce $I_{raw} = 1.7$ mA current draw at a wind speed of 5 m/s. There are also many other similar options available as presented in [38]. These can be tailored to a specific environment to provide a sufficiently reliable power source. A nominal $C_c = 5$ F super capacitor is used, and it can be charged to $V_c = 4.1$ V. An output regulator (with $\eta = 60\%$ efficiency) cuts off when the capacitor is discharged to $V_d = 2.25$ V, so there is $I_{in} = \eta I_{raw} = 1.02$ mA current and $C_s = \eta C_c = 3$ F capacitor storage available to power the node. The node actions and the associated charge gained/consumed are summarised in Table 2. The amount of charge obtained from the energy harvester $Q_C = I_{in} \cdot 1\text{s} = 1.02$ mC per second. The amount of charge dissipated during sleeping (RF module in sleep mode and processor in standby) is $Q_s = (I_s + I_{std}) \cdot 1\text{s} = 0.7$ μC per second. The amount of charge dissipated during transmitting (RF module in TX mode and processor in active) is $Q_{tx} = (I_{tx} + I_{act}) \cdot 1\text{s} \approx 49.84$ mC per second. The amount of charge dissipated during receiving (RF module in RX mode and processor in active) is $Q_{rx} = (I_{rx} + I_{act}) \cdot 1\text{s} \approx 16.64$ mC per second. The amount of charge dissipated during processing only (RF module in sleep mode and processor in active) is $Q_p = (I_s + I_{act}) \cdot 1\text{s} \approx 5.84$ mC per second. For convenience, the charges obtained and dissipated in Table 2 are marked with the "+" and "-" signs respectively.

Table 1
LoRa Transceiver Properties.

Property	Value
Spreading factor	10
Bandwidth	125 kHz
Coding rate	4/5
Data rate	976 bps
Transmitting power	14 dBm
Transmitting current draw I_{tx}	44 mA
Receiving current draw I_{rx}	10.8 mA
Sleep mode current draw I_s	0.2 μA
Frequency	868.1 MHz

Table 2
Charges and Node Actions.

Action	Charge/s
Charging Q_C	+1.02 mC
Sleeping Q_s	-0.7 μC
Transmitting Q_{tx}	-49.84 mC
Receiving/listening Q_{rx}	-16.64 mC
Processing Q_p	-5.84 mC

For a node to transmit for 1 second, it needs to be charged for approximately 49 seconds while asleep (equivalent to a 2% duty cycle). For a node that needs to receive/listen for 1 second, it needs to be charged for approximately 16 seconds (equivalent to a 6% duty cycle). The energy budget indicates that the nodes can easily operate at 1% duty cycle (or below) to meet the regulations.

The ping packet uses a short packet size with 22-byte payload (which has the format described in Fig. 4), 4-byte preamble and 2-byte Cyclic Redundancy Check (CRC). The LoRa calculator [35] shows that $T_{ping} = 297$ ms is required to transmit the ping packet and approximately $T_{ack} = 200$ ms to transmit the ACK. It costs a node $Q_{tx\_ping} = Q_{tx} T_{ping} \approx 14.8$ mC to send a ping and $Q_{rx\_ack} = Q_{rx} T_{ack} \approx 3.33$ mC to receive the ACK in one slot, and its next hop node needs $Q_{rx\_ping} = Q_{rx} T_{ping} \approx 4.94$ mC to receive the ping and $Q_{tx\_ack} = Q_{tx} T_{ack} \approx 9.97$ mC to send the ACK. According to the time needed to send a ping and an ACK, we define the slot length as $T_{slot} = 500$ ms and it costs $Q_{rx\_slot} = Q_{rx} T_{slot} \approx 8.32$ mC to listen during a slot. The most power consuming nodes in the network are the route

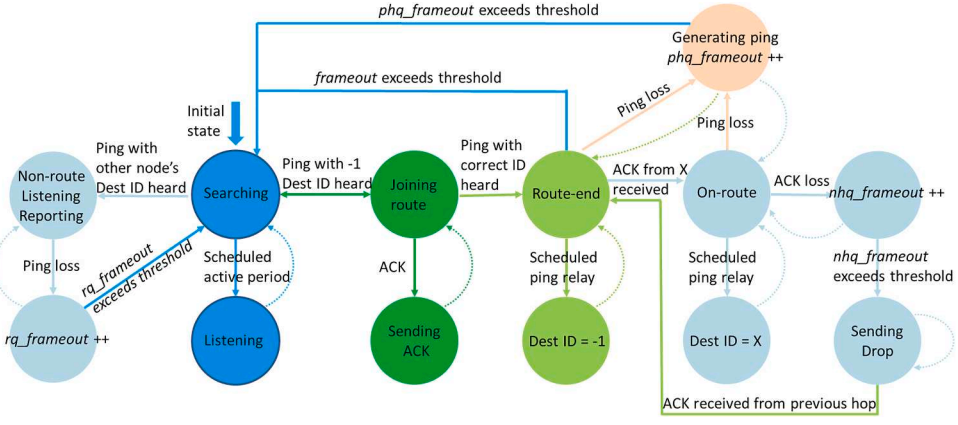


Fig. 10. Updated node operating states.

nodes, which need to be active for at least 3 slots (ping reception, ping transmission, listening/report reception) during one duty cycle. For example, in Fig. 2 the 3 active slots of node D require $Q_{rx\_ping} + Q_{tx\_ack} = 14.91$ mC, $Q_{tx\_ping} + Q_{rx\_ack} = 18.13$ mC and $Q_{rx\_ping} + Q_{tx\_ack} = 14.91$ mC of charge respectively (report packet has the same length as ping). To compensate for the energy consumption during the active period of node D, the amount of charge time required can be obtained as:

$$\frac{2\left(Q_{rx_{ping}} + Q_{tx_{ack}}\right) + Q_{tx_{ping}} + Q_{rx_{ack}}}{Q_C} \approx 47 \text{ s} \quad (1)$$

which makes a maximum affordable 3.2% duty cycle. If we assume node D is active in slot 2 (listening mode) as well, the amount of charge time required becomes:

$$\frac{2\left(Q_{rx_{ping}} + Q_{tx_{ack}}\right) + Q_{tx_{ping}} + Q_{rx_{ack}} + Q_{rx\_slot}}{Q_C} \approx 55.2 \text{ s} \quad (2)$$

which converts to a maximum affordable duty cycle of about 3.6%. As described earlier, we consider each route node to have 1 slot reserved for non-route nodes to send report packets. This number could be further extended given an improved energy budget. However, the bottleneck is the limited payload space in ping packets, so having more frequent reports could be unnecessary. To make sure the energy budgets satisfy the operation of the route node during route formation (no report packets) the amount of charge time required is:

$$\frac{Q_{rx\_ping} + Q_{tx\_ack} + Q_{tx_{ping}} + Q_{rx_{ack}} + 2Q_{rx\_slot}}{Q_C} \approx 48.7 \text{ s} \quad (3)$$

which converts to a maximum affordable duty cycle of about 4.1%. Similarly, the amount of charge time required for a non-route node during normal operation is:

$$\frac{Q_{tx_{ping}} + Q_{rx_{ack}} + 3Q_{rx\_slot}}{Q_C} \approx 42.2 \text{ s} \quad (4)$$

which converts to a maximum affordable duty cycle of about 4.7%. The amount of charge time required for a searching node is:

$$\frac{4Q_{rx\_slot}}{Q_C} \approx 32.6 \text{ s} \quad (5)$$

which converts to a maximum affordable duty cycle of about 6.1%. When selecting the appropriate frame size, the results of Eqs. (1)-(5) should be considered while taking into account the associated energy parameters. To meet the 1% target duty cycle (active) we define all nodes to have a 4-slot (2 seconds) active period per frame, and each frame has 400 slots (200 seconds). So, it takes 100 frames (about 5 and half hours) for a searching node to explore every slot. Simulation results of route formation time with the above parameters will be provided in the following subsections unless further specified. Table 3 summarises the duty cycles of the nodes with different tasks in the network. The maximum affordable active column shows the upper bound of the duty cycle while exploiting all the energy budget. The frame size and active duration can be configured in a flexible way to meet the requirements of the networks with different energy budgets and the RF regulations.

Table 3
Nodes and Duty Cycles.

Node type	Duty cycle (active)	Transmission only	Maximum affordable active
On-route (sending ping and receiving report)	1%	0.35%	3.6%
On-route (sending ping)	1%	0.25%	4.1%
Non-route (sending report)	1%	0.15%	4.7%
Non-route (searching)	1%	0%	6.1%

5.2. Route formation of network with equal node spacing

We first evaluate the route formation capability of IRIS by considering a hypothetical linear pipeline network. The network has 300 nodes including the end base station. The nodes are deployed 500 meters from each neighbour and the total network length is 150 km. We assume that the LoRa devices are set to long-range mode in which they have a 20 km communication range with a line-of-sight (LoS) path (which is a common assumption [39] and the record is 702 km [40]). The first sets of simulations are based on this configuration in order to find an appropriate combination of the IRIS protocol parameters mentioned in Section 4 (*frameout* and *conlimit*). Both parameters have significant impact on IRIS performance. Setting a large *frameout* may cause a route-end node to return to the searching state very late when it has no other nodes in range to become a next hop. However, if *frameout* is too small, the route-end node may miss hitting the active period of a potential next hop node in the search state, given the ultra-low duty cycle. Having a large *conlimit* may create unnecessary hops in the route and increase the end-to-end delay of ping packets. On the other hand, a small *conlimit* decreases the probability of finding a next hop before *frameout* expires which increases the time needed for route formation. Monte Carlo simulations are necessary to discover the impact of different parameter configurations and the sensitivity of the system to the settings.

Fig. 11 shows the Cumulative Distribution Function (CDF) of the time to form a route with different combinations of parameters. Using CDF allows the results to capture the varying performance across simulations of the same parameters due to the randomness in the simulations (e.g., searching nodes start their active periods randomly). The simulation results are used to identify the combination of parameters with better performance. The time to form a route is defined as the time from the instant that the first ping initiated by the base station until the time the other base station receives a ping. Each curve represents the route formation results of 50 simulations which captures the randomness during route formation. This applies to all simulation results that appear later if not specified otherwise. Fig. 11 shows that the route formation is not sensitive to *frameout* when *conlimit* is set to 1. Fig. 11 also shows the simulation results matching the estimated route formation time. The route formation time estimation is provided later in subsection C. Fig. 11 shows that a *conlimit* = 1 is slightly better than larger *conlimit* values.

The first impression is that the combination of *frameout* = 50 and *conlimit* = 1 is better than other settings. Generally, in 90% (or more) of the simulations of any parameter combination, a route can be found

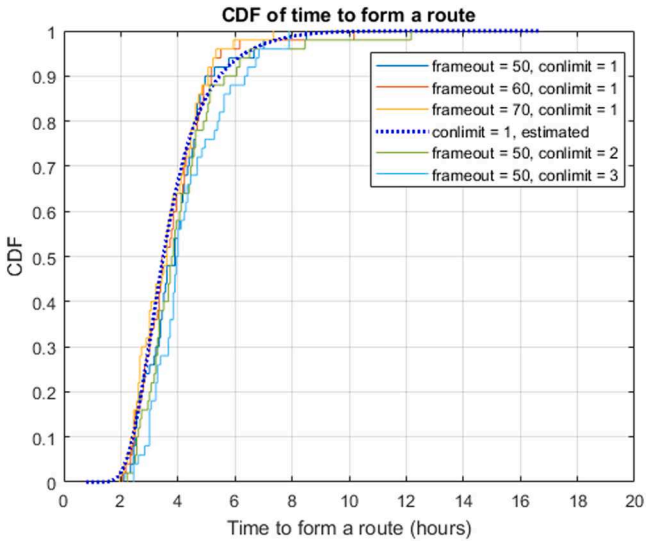


Fig. 11. Results of route formation with different parameters.

within less than 7 hours and the number of hops are within the range from 11 to 13 hops. Once a route is found, the end base station will receive one ping packet which traverses through all route nodes with 22-byte payload every frame. This also indicates that the maximum network throughput is 396 bytes/hour including overheads (27 bits per packet), which is sufficient for long-term monitoring tasks with nodes reporting frequency at the level of hours. Note that the number of hops does not affect the network throughput as long as the number of slots per frame and the number of active slots per frame are fixed. The network throughput is obtained based on the parameters in Table 1 and it can be changed by adjusting the Spreading Factor and bandwidth. E.g., a higher Spreading Factor reduces the throughput but increases range, higher bandwidth increases the throughput but reduces range.

5.3. Route formation time estimation of network with equal node spacing

It is possible to consider an analytical approach to determine some aspects of performance. Many probability notations are involved while estimating the route formation time, and the general rule of thumb is (unless specifically described):

$p(variable)$

denotes the Probability Distribution Function (PDF) or Probability Mass Function (PMF) of the *variable*.

$P_{variable}(condition)$

denotes the CDF of the *variable* when the *condition* is met.

$\mathbb{P}(event\ )$

denotes the probability for the *event* to occur.

$\boldsymbol{P}$

denotes a vector or a matrix.

We consider finding the route nodes as a renewal process, and therefore the time it takes for a route node to find the next route node follows a certain distribution and the distance between two consecutive route nodes follows a certain distribution. The renewal process stops when the last route node is within the coverage range of the end base station.

As described earlier, the example network consists of 300 nodes and the distance between two neighbour nodes is 500 meters. Each node has the communication range of 20 km in radius, and the nodes have 1% duty cycle with 400 slots per frame, and the number of duty cycles (frames) for a searching node to traverse all 400 slots is $n_{dc} = 100$. Now we can look at the process of the base station (the first route node) finding the second route node. The base station emits a ping packet per frame and there are 40 searching nodes within the communication range of the base station. Since the searching nodes are started at random times and the activities for a searching node to listen to different slots are exclusive, in the first frame where the base station emits the first ping, the probability for each searching node (that is within range of the sender) to receive this ping is $\frac{1}{n_{dc}}$, and in the second frame this probability becomes $\frac{1}{n_{dc}-1}$ because the ping is always emitted in the same slot. In the 100th frame this probability becomes 1. We define:

$$P_{pf}(i) = \frac{1}{n_{dc} - i + 1} \tag{6}$$

which is the probability for a node receiving a ping until frame i (without receiving a ping in previous frames), and the probability of no searching nodes receiving the ping from the base station in the first frame is:

$$P_{npr_j}(1) = \left(1 - P_{pf}(1)\right)^{r_j} \tag{7}$$

where $j = 2$ represents the process of finding the second route node and r_j denotes the number of potential searching nodes that could become the j^{th} route node. In the case of the base station trying to find the second route node, $r_2 = 40$ and all 40 nodes have the equal probability of becoming the second route node. The probability of no searching nodes receiving the ping within i frames is:

$$P_{npr_j}(i) = P_{npr_j}(i-1) \times \left(1 - P_{pf}(i)\right)^{r_j},\ i > 1 \tag{8}$$

Using Eqs. (7) and (8) we can calculate the probability for the ping to be received by at least one searching node in exactly the i^{th} frame:

$$P_{pr_j}(i) = \begin{cases} 1 - P_{npr_j}(i), & i = 1 \\ \left(1 - P_{npr_j}(i)\right) - \left(1 - P_{npr_j}(i-1)\right), & i > 1 \end{cases} \tag{9}$$

With Eq. (9) we can obtain the PMF of the time to find the second route node because we know $r_2 = 40$. However, to further move on to the third route node we need to know r_3 which is dependent on the location of the second route node. Let's assume the node 500 meters away from the base station is node 1, the node 1 km away from the base station is node 2…, the node 20 km away from the base station is node 40. Nodes 1 to 40 have the same 1/40 probability of becoming the second route node. If node 1 becomes the second route node, nodes 2 to 40 cannot be the third route node because they can hear the ping from the base station (*conlimit*=1), therefore only node 41 can become the third route node. If node 2 becomes the second route node, only nodes 41 and 42 can become the third route node (each with 50% probability) …. If node 40 becomes the second route node, nodes 41 to 80 can become the third route node (each with 1/40 probability). So r_3 has the equal 1/40 probability to be 1 to 40. We can also obtain the distribution of r_4:

$$\mathbb{P}(r_4 = m) = \sum_{i=41-m}^{40} \frac{1}{40i},\ m = 1 \ldots 40 \tag{10}$$

Then we can obtain the distribution of r_5:

$$\mathbb{P}(r_5 = 40) = \mathbb{P}(r_4 = 40)\cdot\frac{1}{40} + \mathbb{P}(r_4 = 39)\cdot\frac{1}{39} + \cdots + \mathbb{P}(r_4 = 1)\cdot 1 \tag{11}$$

$$\mathbb{P}(r_5 = 39) = \mathbb{P}(r_4 = 40)\cdot\frac{1}{40} + \mathbb{P}(r_4 = 39)\cdot\frac{1}{39} + \ldots + \mathbb{P}(r_4 = 2)\cdot\frac{1}{2} + \mathbb{P}(r_4 = 1)\cdot 0 \tag{12}$$

……

$$\mathbb{P}(r_5 = 1) = \mathbb{P}(r_4 = 40)\cdot\frac{1}{40} + \mathbb{P}(r_4 = 39)\cdot 0 + \ldots + \mathbb{P}(r_4 = 1)\cdot 0 \tag{13}$$

Using Eqs. (10) to (13) we can obtain the distribution of any r_j:

$$\begin{bmatrix} \mathbb{P}(r_j = 40) \\ \mathbb{P}(r_j = 39) \\ \mathbb{P}(r_j = 38) \\ \vdots \\ \mathbb{P}(r_j = 2) \\ \mathbb{P}(r_j = 1) \end{bmatrix}^T = \begin{bmatrix} \mathbb{P}(r_{j-1} = 40) \\ \mathbb{P}(r_{j-1} = 39) \\ \mathbb{P}(r_{j-1} = 38) \\ \vdots \\ \mathbb{P}(r_{j-1} = 2) \\ \mathbb{P}(r_{j-1} = 1) \end{bmatrix}^T \begin{bmatrix} \frac{1}{40} & \frac{1}{40} & \frac{1}{40} & \cdots & \frac{1}{40} & \frac{1}{40} \\ \frac{1}{39} & \frac{1}{39} & \frac{1}{39} & \cdots & \frac{1}{39} & 0 \\ \frac{1}{38} & \frac{1}{38} & \frac{1}{38} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{1}{2} & \frac{1}{2} & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \tag{14}$$

which can be simplified as:

$$\boldsymbol{P}(r_j)^T = \boldsymbol{P}(r_{j-1})^T \boldsymbol{R} \tag{15}$$

And we know all elements of $\boldsymbol{P}(r_3)$ equal to 1/40. $\boldsymbol{P}(r_j)$ also represents the PMF of distance between route nodes $j-1$ and $j-2$, because the distance is the same 500 meters between two neighbour nodes. Therefore, we can obtain the distribution of the distance between the base station and the j^{th} route node:

$$\boldsymbol{P}(\ D_j) = \boldsymbol{P}(\ r_3) * \boldsymbol{P}(\ r_4) * \ldots * \boldsymbol{P}(\ r_{j+1}) \tag{16}$$

where $*$ is the operator of convolution and D_j is the distance between the base station and the j^{th} route node.

Next, we need to obtain the distribution of the number of route nodes required to form a network over the 150 km length. For example, the probability of j route nodes forming the 150 km network equals to the probability of j route nodes forming a network longer than 150 km while $j-1$ route nodes forming a network shorter than 150 km.

$$\begin{aligned}\mathbb{P}(nr=j) &= \mathbb{P}\big(D_j \geq 150km \cap D_{j-1} < 150km\big) \\ &= \sum_{x=150km}^{\infty} \mathbb{P}(\ D_j = x) - \sum_{x=150km}^{\infty} \mathbb{P}(\ D_{j-1} = x)\end{aligned} \tag{17}$$

The values of $\mathbb{P}(\ D_j = x)$ can be obtained from elements of $\boldsymbol{P}(\ D_j)$. Combining Eqs. (9) and (14) we can obtain the probability of the ping packet received by at least 1 searching node across all potential r_j searching nodes in every possible i^{th} frame:

$$\boldsymbol{P}_p(i, r_j) = \begin{bmatrix} P_{p40}(1)\mathbb{P}(\ r_j = 40) & P_{p39}(1)\mathbb{P}(\ r_j = 39) & \cdots & P_{p1}(1)\mathbb{P}(\ r_j = 1) \\ P_{p40}(2)\mathbb{P}(\ r_j = 40) & P_{p39}(2)\mathbb{P}(\ r_j = 39) & \cdots & P_{p1}(2)\mathbb{P}(\ r_j = 1) \\ \vdots & \vdots & \ddots & \vdots \\ P_{p40}(100)\mathbb{P}(\ r_j = 40) & P_{p39}(100)\mathbb{P}(\ r_j = 39) & \cdots & P_{p1}(100)\mathbb{P}(\ r_j = 1) \end{bmatrix} \tag{18}$$

which is a 100×40 matrix because $r_j \in [1, 40]$ and $i \in [1, 100]$. Now we can obtain the combined distribution of the network distance and route formation time with j route nodes:

$$\boldsymbol{P}(\ DT_{r_j}) = \boldsymbol{P}_p(i, r_2) * \boldsymbol{P}_p(i, r_3) * \ldots * \boldsymbol{P}_p(i, r_j) \tag{19}$$

where the columns of the matrix represent the network distance and the rows represent the route formation time. Summing all elements in each row gives the probabilities of finding the route in different numbers of frames, given the condition of having j route nodes in the network:

$$\boldsymbol{P}(\ T_{r_j}) = \boldsymbol{P}(\ DT_{r_j})\mathbf{1} \tag{20}$$

where $\mathbf{1}$ is a column of ones. Then multiply the matrix with the probability of having j route nodes in the network and sum the results across all possible j to obtain the distribution of network formation time:

$$\sum_j \boldsymbol{P}(\ T_{r_j})\mathbb{P}(nr=j) \tag{21}$$

Using Eq. (21) we can easily obtain the CDF of network formation time shown in Fig. 11.

5.4. Route formation of networks with random node spacing

The performance stability of IRIS has been tested with different network topologies. First we construct a random linear pipeline network with 300 nodes. The network is generated node-by-node, with 80% of the neighbour node distances under 2 km and 20% of the neighbour node distances between 2 km and 5 km. The distance between two neighbour nodes follows the uniform distribution. The total length of this random network is approximately 467 km. Fig. 12 shows that the route formation is faster when *conlimit* is set to 1, which is consistent with earlier simulation results. Fig. 12 also shows that setting *frameout* = 50 slightly outperforms other *frameout* values, which is also consistent with the first impressions we have earlier. With the best parameter configuration, a route is found within 35 hours for 90% of the simulations, and 50 hours for all simulations. Given the time scale of typical WSN deployments in these long-range monitoring scenarios, a 2-day route formation phase is negligible to a WSN which will operate for months or years. The average number of hops of all found routes is about 35, which is much larger than the previous hypothetical network (because of the longer distance between neighbour nodes).

We also construct two other random linear networks following the same rules to make sure the route finding process operate correctly with different network topology configurations. Fig. 13 shows the distance from the different numbers of nodes to the start base station of the three generated networks. The networks have the total length of 467 km, 423 km and 430 km respectively. Fig. 14 compares the route formation time of the three networks and the estimated route formation time for the three networks are included to match the simulation results. The estimation is provided later this subsection E. The average numbers of hops are 35, 31 and 32 for all 3 networks. These also indicate that a ping packet carrying monitoring data can traverse through the 3 networks every 35, 31 and 32 seconds to provide frequent updates for the monitoring tasks. As expected, IRIS can form routes faster for networks with fewer hops and shorter lengths. The route formation time for 90% of the

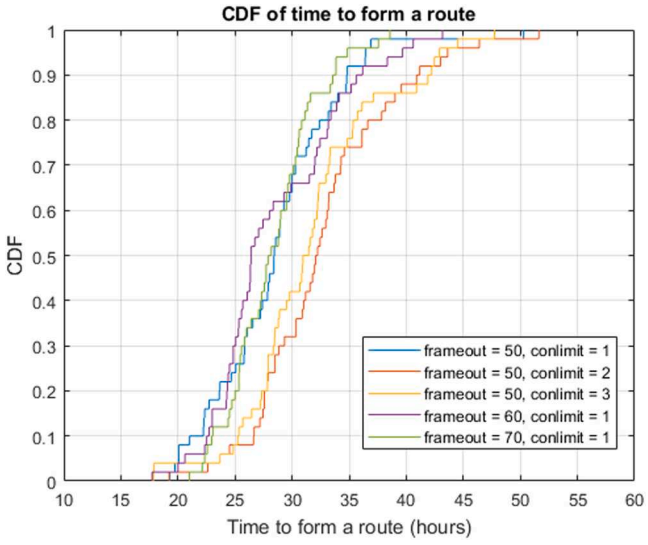


Fig. 12. Results of route formation with different parameters (random network 1).

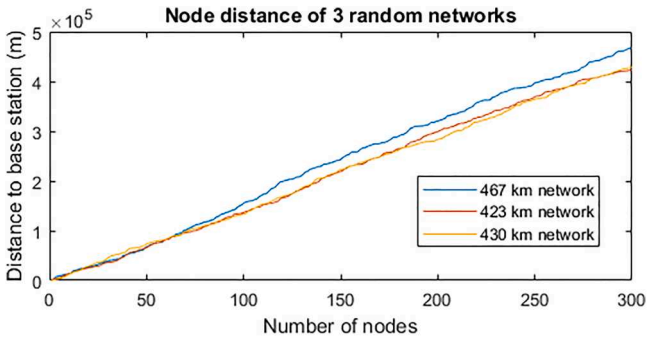


Fig. 13. Node distance of randomly generated networks.

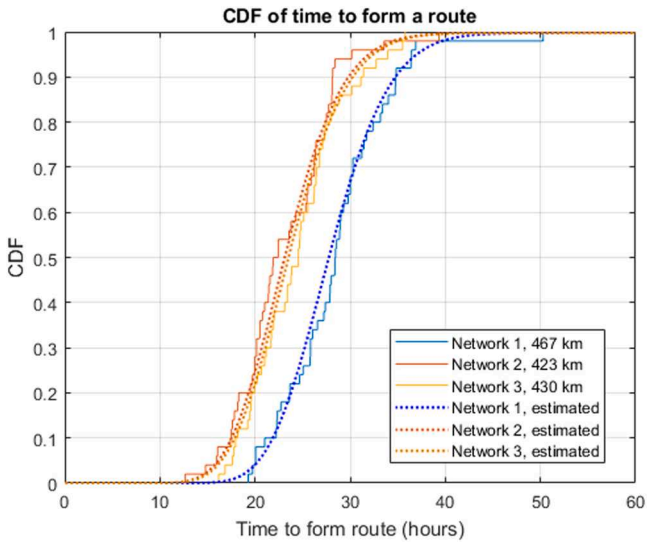


Fig. 14. Results of 3 random networks.

simulations are 35, 28 and 31 hours for 3 networks respectively. IRIS is able to provide reliable route formation for different pipeline networks without any prior topology information while consistently maintaining low energy cost, which is the major advantage of the protocol.

5.5. Route formation time estimation of networks with Random node spacing

As described in earlier, the random networks are generated node by node with 80% of the neighbour node distances under 2 km and 20% of the neighbour node distances between 2 km and 5 km and the neighbour node distances follow a uniform distribution. So, the PDF of the distance x between two consecutive nodes is:

$$f(x) = \begin{cases} \frac{2}{5}, \ for\ x \in (0,\ 2]\ km \\ \frac{1}{15}, \ for\ x \in (2,\ 5]\ km \\ 0, \ otherwise \end{cases} \tag{22}$$

Assuming another variable y follows the same distribution, and the PDF of the variable $z = x + y$ (which is the distance across the subsection of a network containing 2 consecutively generated nodes) can be calculated by:

$$f_z(z) = \int_{-\infty}^{\infty} f_x(x) f_y(z-x) dx \tag{23}$$

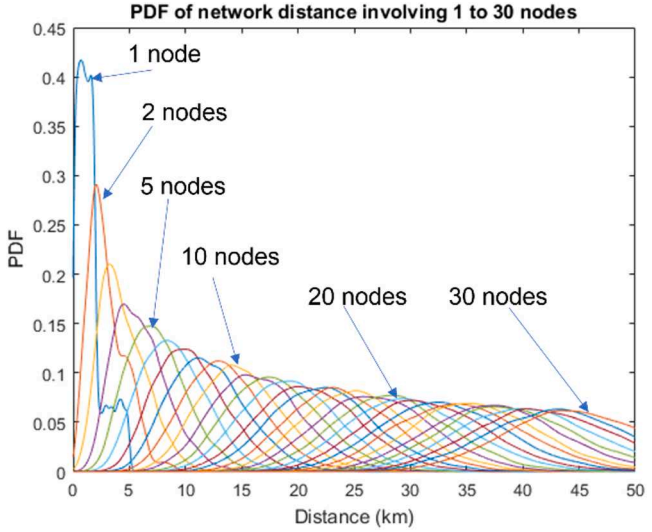


Fig. 15. PDF of network distance involving different numbers of nodes.

With more nodes involved in the network subsection, we need to cascade the integral many times and obtaining the analytical expression of the PDF of the distance of the network subsection is only theoretically possible. However, with Matlab, we can obtain the estimated PDF by generating massive samples and fit the samples to a certain distribution. Fig. 15 shows the estimated PDF of network distance involving different numbers of nodes (assuming there is already one base station at the start of the network), and we can see that it is extremely unlikely for the network to be shorter than 20 km when there are 23 or more nodes in the network. We defined this PDF as $p_{ndm}(d)$, where the number of nodes $m \in [1,\ 30]$. In Matlab $p_{ndm}(d)$ is in the form of discrete PMF with a sample at every meter from 1 meter to 50 km, and this PMF form will be used later in the analysis.

Similarly, we start the analysis by finding the distribution of the distance between the base station (first route node) and the second route node. In this case all searching nodes within the 20 km communication range of the base station could be the second route node. So, the probability of the number of searching nodes within 20 km of the base station can be obtained as:

$$\mathbb{P}(m|D<20km) = P_{ndm}(D<20km) - P_{nd(m+1)}(D<20km) \tag{24}$$

where $m \in [1,\ 29]$, and $P_{ndm}(D<20km)$ is the CDF of $p_{ndm}(d)$ with the condition of network distance shorter than 20 km. Each of the m searching nodes has the same probability $(1/m)$ to become the second route. The distribution of the distance between the base station and the second route node is a mixture distribution with the weights of $\mathbb{P}(m|D<20km)$, multiplied with the PMF of the distance between the base station and the second route with m searching nodes involved, which can be further break down as another mixture distribution with weights of $1/m$, multiplied with the conditional PMF of the network distance with 1 to m nodes involved given the condition of the network distance being shorter than 20 km:

$$p_{dr_2}(d) = \sum_{m=1}^{30} \mathbb{P}(m|D<20km) \frac{\sum_{j=1}^{m} p_{ndj}(\ d|d<20km)}{m} \tag{25}$$

where

$$p_{ndj}(d|d<20km) = \begin{cases} p_{ndj}(d)/P_{ndj}(d<20km), \ if\ d<20km \\ 0, \ if\ d \geq 20km \end{cases} \tag{26}$$

which can be computed from the PMF in Fig. 15.

In terms of finding the third route node, not all searching nodes within the 20 km communication range of the second route node can become the third route node since *conlimit*=1 and some searching node can hear the ping emitted by the base station. The number of nodes that can become the third route node depends on the location of the second rout. For example, if the second route node is 2 km away from the base station, the nodes that are 2 km to 20 km away from the base station cannot become the third route node and only the nodes that are 20 km to 22 km (a 2 km network subsection) away from the base station can become the third route node. To compute the distribution of the distance between the second and the third route nodes we need to evolve Eq. (25) by making another mixture distribution on top with the weights of probability of the distance between base station and the second route node:

$$p_{dr_3}(d) = \sum_{x} P_{dr_2}(d=x) \sum_{m=1}^{30} \mathbb{P}(m|D<x) \frac{\sum_{j=1}^{m} p_{ndj}(\ d|D<x)}{m} \tag{27}$$

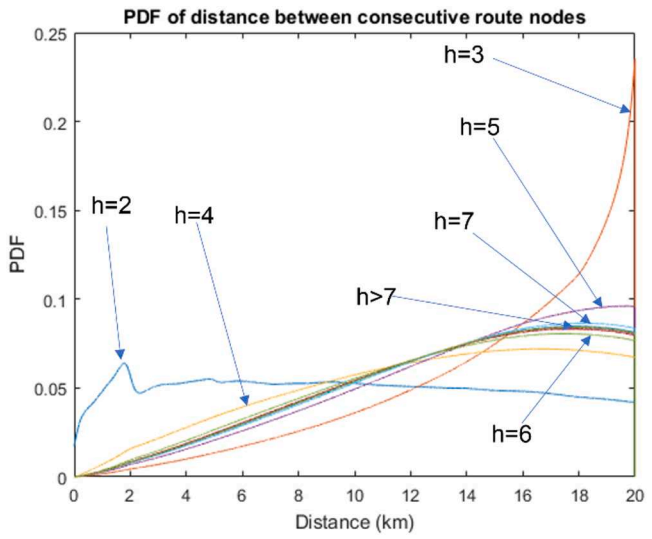


Fig. 16. PDF of distance between two consecutive route nodes.

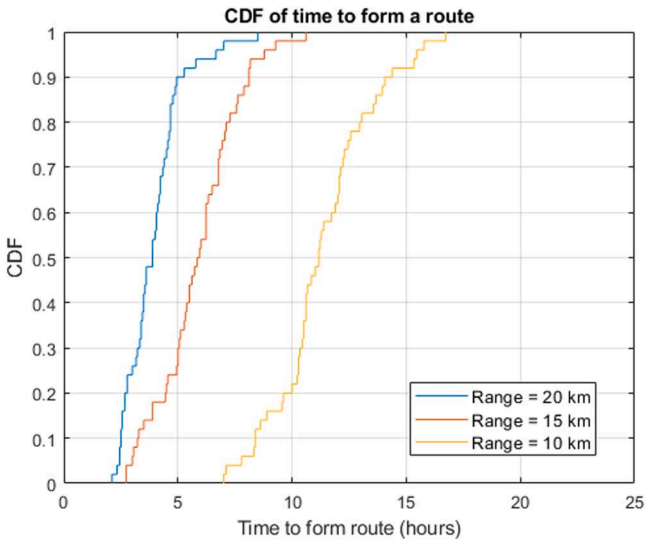


Fig. 17. Hypothetical network with different communication range.

where $P_{dr_2}(d=x)$ is the exact probability of the distance between base station and the second route node equals to x which can be obtained from the discrete PMF $p_{dr_2}(d)$. And x is the sample of the distance, which has the range from 1 meter to 20 km (one sample every meter) since the distance between the base station and the second route node cannot be larger than 20 km. With Eq. (27) we can further compute the distribution of the distance between the h^{th} and $h-1^{th}$ route node:

$$p_{dr_h}(d) = \sum_x P_{dr_{h-1}}(d=x) \sum_{m=1}^{30} \mathbb{P}(m|D<x) \frac{\sum_{j=1}^{m} p_{ndj}(\ d|D<x)}{m} \tag{28}$$

Fig. 16 shows the computed $p_{dr_h}(d)$ with different h, and we can see that the distributions become very similar when $h > 7$.

Similar to Eq. (16) in the previous subsection, we can convolve the PMF of $p_{dr_h}(d)$ to obtain the distribution of the network distance involving h route nodes:

$$\boldsymbol{P}(\ D_h) = p_{dr_2}(d) * p_{dr_3}(d) * \ldots * p_{dr_h}(d) \tag{29}$$

where D_h is the network distance involving h route nodes.

From here the analysis merges with the analysis in subsection C Eqs. (16) to ((21)). We can compute the probability of having different numbers of route nodes given the condition of a certain network length:

$$\mathbb{P}(nr=h) = \mathbb{P}(D_h \geq X \cap D_{h-1} < X\ ) = \sum_{x=X}^{\infty} \mathbb{P}(\ D_h=x) - \sum_{x=X}^{\infty} \mathbb{P}(\ D_{h-1}=x) \tag{30}$$

There are three random networks used in the simulations and the values of X are 467, 423 and 430 km.

Then we can compute the probability of the number of searching nodes that could become the h^{th} route node:

$$\mathbb{P}(\ r_h=m) = \sum_x P_{dr_{h-1}}(d=x)\mathbb{P}(m|D<x) \tag{31}$$

where $\mathbb{P}(m|D<x)$ can be computed with Eq. (24). Since we generated $p_{ndm}(d)$ earlier for up to 30 nodes, we can compute $\mathbb{P}(\ r_h=m)$ for $m \in [1, 29]$. Now we can construct the matrix $\boldsymbol{P}_p(i,r_j)$ using Eq. (18) and the matrix has the dimension of 100×29. Then following Eqs. (19) to (21) we can obtain the CDF of the route formation time for all three random networks with *conlimit* = 1 (Fig. 14).

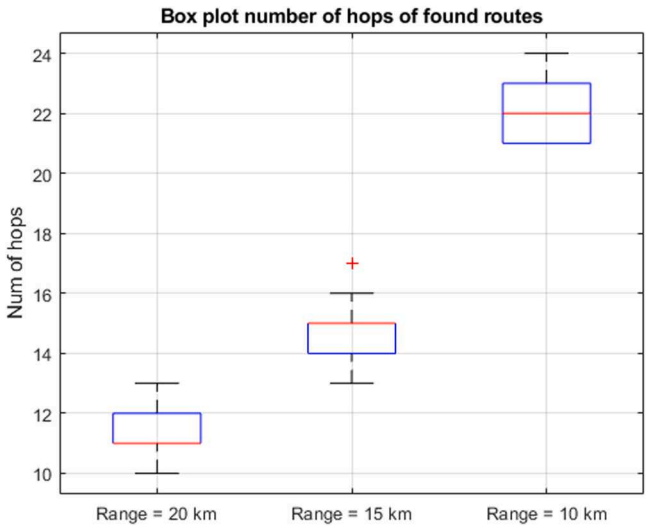


Fig. 18. Hypothetical network with different communication range.

5.6. Route formation with different wireless coverage

When a LoRa transceiver fails to reach the 20 km communication range this could be due to many different reasons, IRIS can adapt to the reduced range and find routes with a greater number of hops. Fig. 17 shows the route formation time of the hypothetical network (500 m neighbour distance) with different communication ranges. We can see that the 90th percentile point is increased to 8 and 14 hours with the reduced 15 km and 10 km range respectively.

Fig. 18 shows a boxplot of numbers of hops of all found routes, where the upper and lower bounds of box represent 75% and 25% of the samples with the red lines as the median value 50th percentile (the 10 km plot has no lower bound whisker because it overlaps with the box). The two whiskers of each box are the upper and lower 1.5 interquartile ranges. The red plus sign represents an outlier outside the upper and lower 1.5 interquartile ranges. When the communication range is reduced by half (10 km), the average number of hops is doubled. The route formation time is almost three times as the 20 km range, because number of potential next hop nodes is reduced, and the route-end node

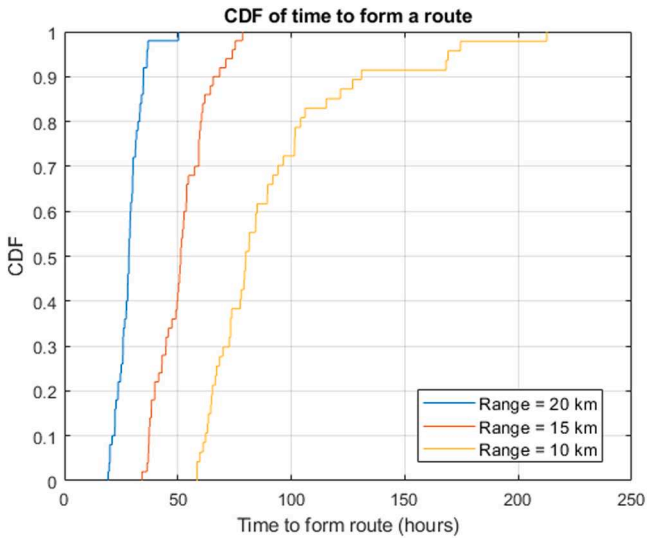


Fig. 19. Random network 1 with different communication range.

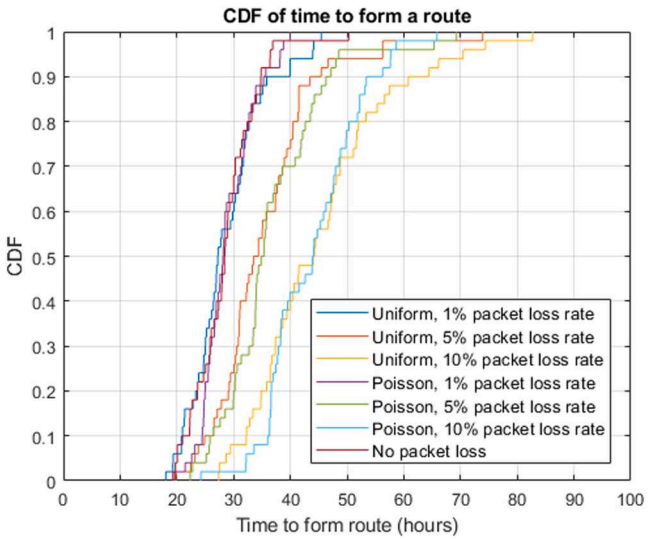


Fig. 21. Route formation time with packet losses.

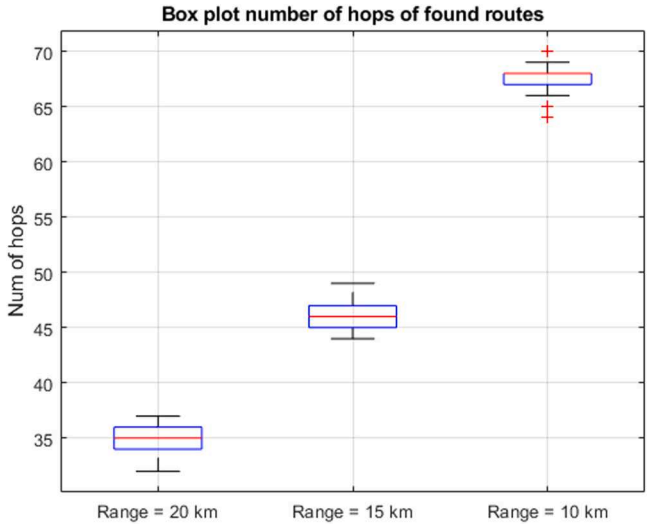


Fig. 20. Random network 1 with different communication range.

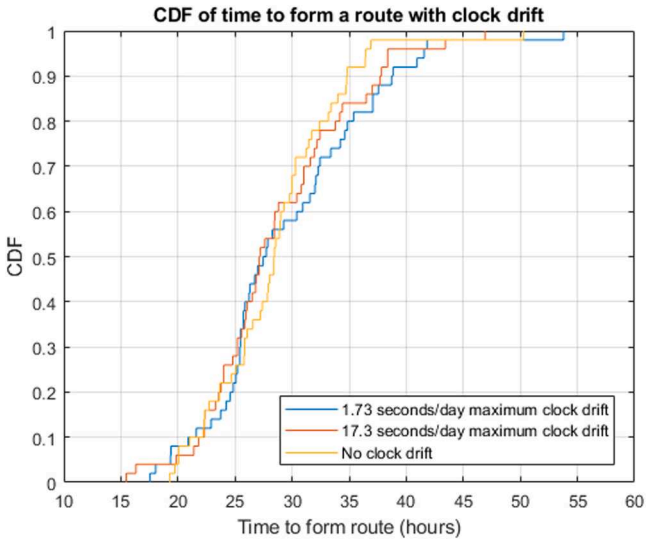


Fig. 22. Route formation time with clock drift.

has to wait longer for them to tune their active periods to receive the ping. Figs. 19 and 20 show the same results of the first random network (random network 1 in Fig. 12) constructed. Similar increasing trends of the route formation time and number of hops can be observed with the reduced communication range. When the communication range is halved to 10 km, it takes 4 times longer to find a route because of much less potential next hop nodes in range. The number of hops to reach the end base station is also almost doubled. These results show the consistently guaranteed route formation with different communication ranges.

5.7. Robustness against unreliable connections and devices

In previous simulations we assume that all nodes have perfect wireless connections and all packets sent during the active periods of the destination nodes are received successfully. To evaluate how the unreliable connections and packet losses affect the IRIS performance, we simulate the route formation process with different packet loss rates. The times between two successive packet loss events are configured to follow either uniform or exponential distributions (Poisson process). In these simulations, we configure the parameters related to improving the robustness against packet losses as follow. *nhq_frameout* is set to 10, *phq_frameout* is set to 50 and *rq_frameout* is set to 10. From Fig. 21 we can see that the simulations with the same packet loss rate of both distributions have similar performance.

1% packet losses have a negligible effect on the route formation time. 90% of the simulations can find routes within 60 hours even with 10% packet losses, compared with the 35 hours of 0% packet loss. Note that the packet loss rate is applied to all packet types and some actions require two packets to complete (e.g., joining the route).

IRIS requires loose time synchronisation to align the active periods of nodes with the associated ping packets. Packet collision is rare due to the unique nature of the actions. Clock drift is a common effect of low-cost oscillators which will cause the timing of slots to slowly shift across different nodes. LoRa devices could have 20 ppm up to maximum 200 ppm clock drift [35], which are 1.73 and 17.3 seconds/day or 4 and 40 ms/frame. The slot length is 500 ms so if a node synchronises its clock to the ping packets every frame and starts its active period slightly earlier with a 50 ms guard period (10% slot duration), the effect of clock drift should be compensated. In Fig. 22 we can see that clock drift only

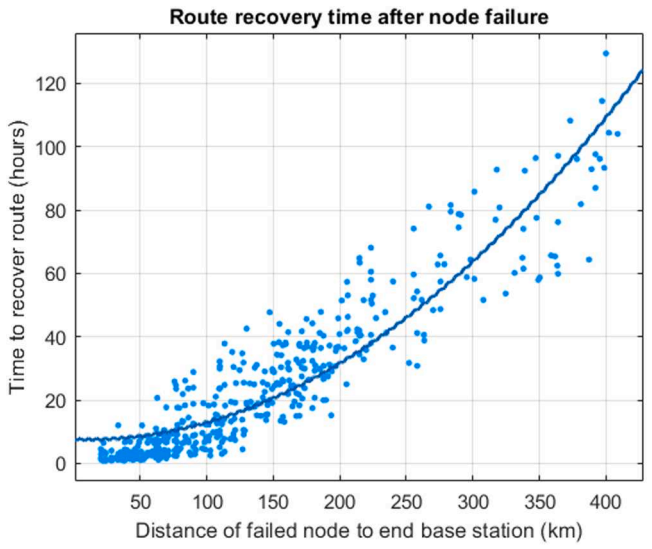


Fig. 23. Route recovery time from node failure.

slightly affects the route formation time.

5.8. Route recovery from node failure

Node failure is common during the operation of WSNs given the nature of their tasks. Failure of non-route nodes will not cause significant impact to the operation of IRIS, however if a route node stops functioning, IRIS must establish a new route from where the route breaks. As described earlier in Section 3 and Section 4, the previous hop of the failed node becomes a route-end node and starts to look for a node to join the route. To evaluate how IRIS recovers from route node failure, we simulate a random route node failure after a route is found and record the time that a new route is established. Fig. 23 shows the route recovery time of 494 node failures simulated with 90 randomly found routes of all three randomly constructed networks (30 routes per network). A clear relationship between the route recovery time and the distance of the failed node to the end base station can be observed. It takes a longer time for the route to recover if the failed node is far away from the end base station, which is almost equivalent to rebuilding the route.

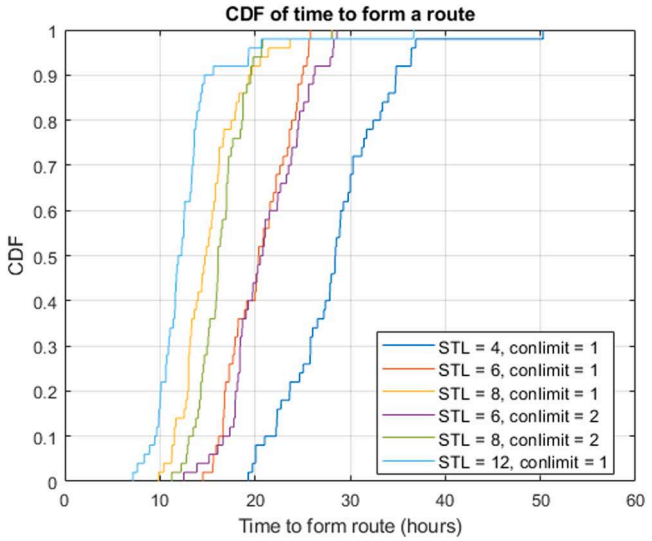


Fig. 24. Faster route formation time.

5.9. Route formation time and duty cycle

The IRIS protocol is designed to operate with flexible duty cycle configurations which could be affected by both spectrum regulations and node energy budget. For example, the previous simulations targeting 1% duty cycle (active) have not exploited all energy budget (affordable duty cycle configurations are shown in Table 3) provided by the energy harvesting device. Fig. 24 shows the route formation time of different slots-to-listen (STL) values per frame. Significant route formation time improvements can be observed when using larger STL (the number of hops is similar across different STL values). When exploiting more energy budget (e.g., when STL is set to 12) 90% simulations only require 15 hours to find routes, which is 270 frames. On the other hand, IRIS can support devices with lower energy budget by tuning down the duty cycle, resulting longer route formation time. The flexibility of duty cycle configurations allows the IRIS protocol to be applicable to networks with different energy budgets and spectrum regulations.

5.10. Normal operation stage after route formation

Once the route formation is completed the Link ID field in the ping packet is changed to "1" by the base station and the non-route nodes are allowed to send report packets to the route nodes that they hear the ping packets from. One-hop clusters are formed around each route node and the non-route nodes that report to the route node become members of the cluster. The non-route nodes need to contend to access the shared resource to send their report packets periodically and there are different resource allocation methods in the literatures that can be adapted here to manage the shared resource and congestion.

Conventional handshake mechanisms such as Carrier Sense Multiple Access – Collision Avoidance (CSMA/CA) are able to avoid the packet collisions, but they require additional overheads and increase the duty cycle. A similar but simplified optimisation method [41] for the resource allocation of the reserved slots across multiple frames can be adapted to reduce the collisions between the report packets sent by non-route nodes. Machine learning [42] can be applied to the non-route nodes to improve the performance of the network after route formation without overheads and reduce collisions. The balance between network throughput, energy consumption and service reliability can also be optimised by Machine Learning [43] to improve the steady-state performance. The contract-based incentive mechanism [44] can be applied to improve the fairness of the report frequencies of the non-route nodes and smooth the access demands. The access class barring (ACB) [44] can be used by the route nodes to prevent the payload space from being filled too early while traversing the route. The request-response mechanism [45] can be implemented to allow the base station to request extra reports from certain nodes on demand without having to wait for the periodical report. The congestion control proposed in [46] is able to manage the level of contention across neighbour clusters and allow the non-route nodes to report to less congested route nodes that the

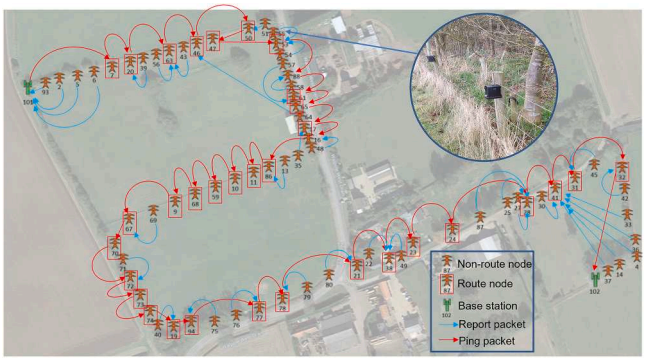


Fig. 25. Deployed test network across a farm and example of one formed route.

non-route nodes can receive ping packets from.

6. Implementation and evaluation

To evaluate the capability of IRIS, we implemented a linear network including 71 attenuated LoRa nodes and 2 base stations installed across the fence line of a farm with the total distance of about 1.3 km. In this section we present the details of the implemented network for testing IRIS and discuss the observed network performance results during long term operation.

6.1. Implementation details

The LoRa nodes we used for the test network were Murata CMWX1ZZABZ-078 modules [47] and the base stations were based on the RAK2245 modules [48]. The Murata modules were powered by 2 regular C cell batteries. Fig. 25 shows the locations of the deployed test network overlayed on a satellite map. The green nodes (101 and 102) were base stations, and the orange nodes were LoRa nodes. Due to the short distance between successive nodes (mostly between 10 and 20 meters), 20 to 30 dB of attenuation was applied to all nodes to emulate a much longer node spacing (usually over 1 km) in practical scenarios. With the attenuation the typical communication range of a node became 30 to 50 meters. Note that the PCB antenna equipped on the Murata modules were not omni-directional, therefore they may have various ranges for different orientations.

Several adjustments were made to the network parameters after observing the performance of the test network. We found that the nodes with the Spreading Factor reduced to 8 (from 10) also provided enough coverage in the test environment. This change allowed the ping packet to carry a larger 50-byte payload which was utilised for more network diagnostic information to be sent to the base station. This also resulted in the configurations of 420 ms slot length and 250 slots per frame (105 seconds per frame). With the new configurations the ping packet length T_{ping} became 245 ms, the ACK packet length T_{ack} became 122 ms, therefore the transmission only duty cycle of a route node became 0.35% (sending a ping and an ACK for the ping from the previous hop) or 0.47% (an extra ACK for a report) which were still within the 1% duty cycle limit of the regulation. The total active slots per frame was still 4 across all nodes, therefore resulting an active duty cycle of 1.6%. Following these adjustments, *conlimit* was set to 2 and *frameout* was set to 30.

6.2. Performance observations

To allow monitoring and diagnostic of the network, the payload in the report and ping packets were used to carry information such as node battery level, temperature, and the clock drift per frame (the time offset between expected time to receive a ping and the actual time that a ping was received). A spare byte in the ACK (for the ping) was also used to send the ID of the current route-end node back to the starting base station so that the progress of the route formation can be monitored before a route was formed. Fig. 25 shows and example of one of the formed routes. The LoRa nodes marked with a red rectangle were the route nodes and the rest LoRa nodes were non-route nodes. The blue arrows were the report transmissions, and the red arrows were ping transmissions. This example route involved 35 route nodes and 36 non-route nodes.

Unlike the results presented in Section 5 where we simulated massive amounts of route formations, the experiments in a real-world environment had to overcome various practical issues. Due to the heavily attenuated LoRa nodes the link budget was quite limited, therefore changes in the environment could significantly affect the link quality. Note that the test site was in a populated area, therefore activities such as wandering animals, swinging fence due to gusty wind, parked cars, wet long grass / trees, and interference from other wireless devices could all affect the reliability of the network (e.g., blocking LoS). Our

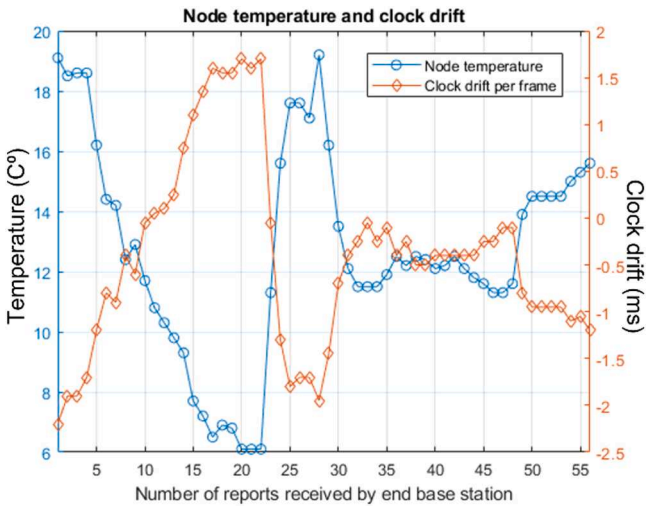


Fig. 26. Node temperature and clock drift per frame.

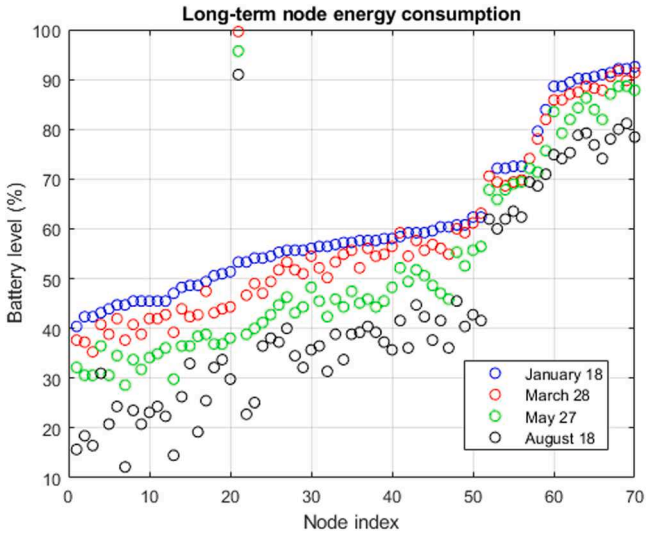


Fig. 27. Long term node energy consumption.

observation showed that when the environment was relatively quiet (e.g., during a quiet, dry night), there were around 2 to 5 lost pings per hour (out of about 35 pings per frame, 34 frames per hour). However, the ping losses could be significantly more when the environment was busy and sometimes causing the route to be reformed. The typical time to form a route from scratch was around 1 to 3 days and the time a route recovered from node failures was around 1 to 2 days, both of which were consistent to the simulation results in Section 5 with a similar number of hops.

Fig. 26 shows a section of the node temperature and clock drift per frame extracted from the report packets received by the end base station. A clear relationship between the two node metrics can be observed. The nodes had the least clock drift when the temperature was about 11.5 C°. This also shows that the clock drift was between ±2.5 ms per frame (frame length is 105 seconds) which was equivalent to about 24 ppm. According to Fig. 22 which showed that IRIS worked well even with 200 ppm clock drift, this 24-ppm clock drift we observed in practice would not affect the network performance.

Fig. 27 shows the long-term node energy consumption from January to August 2022. For the ease of observation, the data was sorted in ascending order according to the battery levels on the January 18th, and one of the nodes had a battery replacement during March. The maximum energy consumption during this 7-month period was 35.5%

and the minimum was 4.7%, and about 60 nodes had the energy consumption between 10% and 25%. This indicates that even with the maximum energy consumption, a node will remain operational for an over 18-month period without battery replacement. Once an energy harvesting device is integrated, a node will have a much longer lifetime.

7. Conclusion

This paper has presented IRIS, a unique low duty cycle cross-layer protocol designed for long-range WSNs with extremely limited power budget. Periodic ping packets has been used to discover and set up a route to an end base station as well as to carry monitoring information. The main features of the IRIS protocol include its simplicity, the consistent low energy cost across all operating conditions and robustness against unreliable wireless connections and devices. The major contribution which makes the IRIS advances the state-of-the-art is that the coordination between MAC and network layers of IRIS does not require prior network information or node geolocation to provide route formation and data delivery with flexible duty cycle configurations. A large number of Monte Carlo simulations have been conducted to find the appropriate parameters to operate the IRIS protocol. An analytical model has been proposed to estimate the route formation time of different networks to validate the simulation results. Simulations results have shown guaranteed route formation under various conditions and robust route recovery after route node failures. A network of 71 attenuated LoRa nodes has been implemented and the capability of IRIS has been validated through long-term operation of the network. Future work includes implementing MAC layer options for normal operation stage and evaluate the network performance.

CRediT authorship contribution statement

Yi Chu: Methodology, Software, Formal analysis, Resources, Writing – original draft. **Paul Mitchell:** Conceptualization, Writing – review & editing, Supervision, Investigation. **David Grace:** Writing – review & editing, Supervision, Investigation. **Jonathan Roberts:** Conceptualization, Software, Validation, Funding acquisition. **Dominic White:** Software, Validation, Funding acquisition. **Tautvydas Mickus:** Methodology, Software, Resources.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Yi Chu reports financial support was provided by TE Connectivity Ltd. Yi Chu has patent Communication network for monitoring a chain based network pending to Tyco Electronics UK Ltd.

Data availability

No data was used for the research described in the article.

Acknowledgment

Monte Carlo simulations were conducted by the York Advanced Research Computing Cluster, which provided the platform of executing massive number of MATLAB simulations in parallel.

References

[1] I.F. Akyildiz, Y. Sankarasubramaniam, E. Cayirci, W. Su, Wireless sensor networks: a survey, Comput. Netw. Chem. Lab., Symp. 40 (2002) 393–422, https://doi.org/10.1016/S1389-1286(01)00302-4. Available DOI:.

[2] G. Werner-Allen, K. Lorincz, M. Welsh, M.Ruiz Omar, K. Lees, Deploying a wireless sensor network on an active volcano, IEEE Internet Comput. 10 (2006) 18–25, https://doi.org/10.1109/MIC.2006.26. Available DOI:.

[3] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, J. Anderson, Wireless Sensor Networks for Habitat Monitoring, WSNA'02, Atlanta, USA, 2002, pp. 88–97, https://doi.org/10.1145/570738.570751. Available DOI:.

[4] J. G. T. Anderson. "Pilot survey of mid-coast Maine seabird colonies: an evaluation of techniques". Bangor, ME, 1995. Report to the State of Maine Dept. of Inland Fisheries and Wildlife.

[5] Mohammad Alex, Simon Carlsen, Stig Petersen, Applications of wireless sensor networks in the oil, gas and resources industries, in: 2010 International Conference on Advanced Information Networking and Applications, 2010, pp. 941–948, https://doi.org/10.1109/AINA.2010.18. Available DOI:.

[6] Rim Negra, Imen Jemili, Abdelfettah Belghith, Wireless body area networks: applications and technologies, Procedia Comput. Sci. 83 (2016) 1274–1281, https://doi.org/10.1016/j.procs.2016.04.266. Available DOI:.

[7] Bushra Rashid, Mubashir Husain Rehmani, Applications of wireless sensor networks for urban areas: a survey, J. Netw. Comput. Appl. 60 (2016) 192–219, https://doi.org/10.1016/j.jnca.2015.09.008. Available DOI:.

[8] Etimad Fadel, et al., A survey on wireless sensor networks for smart grid, Comput. Commun. 71 (2015) 22–33, https://doi.org/10.1016/j.comcom.2015.09.006. Available DOI:.

[9] Tamoghna Ojha, Sudip Misra, Narendra Singh Raghuwanshi, Wireless sensor networks for agriculture: the state-of-the-art in practice and future challenges, Comput. Electron. Agric. 118 (2015) 66–84, https://doi.org/10.1016/j.compag.2015.08.011. Available DOI:.

[10] D.E. Boubiche, et al., Advanced industrial wireless sensor networks and intelligent IoT, in: IEEE Communications Magazine 56, 2018, pp. 14–15, https://doi.org/10.1109/MCOM.2018.8291108. Feb.Available DOI:.

[11] Kofi Sarpong Adu-Manu, et al., Energy-harvesting wireless sensor networks (EH-WSNs): a review, ACM Trans. Sensor Netw. (TOSN) 14 (2) (2018) 10, https://doi.org/10.1145/3183338. Available DOI:.

[12] I. Demirkol, C. Ersoy, F. Alagoz, MAC protocols for wireless sensor networks: a survey, Commun. Mag. 44 (2006) 115–121, https://doi.org/10.1109/MCOM.2006.1632658. Available DOI:.

[13] Y. Wei, J. Heidemann, D. Estrin, Medium access control with coordinated adaptive sleeping for wireless sensor networks, IEEE/ACM Trans. Netw. 12 (3) (2004) 493–506, https://doi.org/10.1109/TNET.2004.828953. Available DOI:.

[14] I. Rhee, A. Warrier, M. Aia, J. Min, M.L. Sichitiu, Z-MAC: a hybrid mac for wireless sensor networks, IEEE/ACM Trans. Netw. 16 (2008) 511–524, https://doi.org/10.1109/TNET.2007.900704. Available DOI:.

[15] IR 2030 – UK interface requirements 2030, available: https://www.ofcom.org.uk/_data/assets/pdf_file/0028/84970/ir-2030.pdf.

[16] LoRaWAN specification v1.1. Available at https://lora-alliance.org/resource_hub/lorawan-specification-v1-1/. Accessed on January 2022.

[17] Wael Ayoub, et al., Internet of mobile things: overview of lorawan, dash7, and nb-iot in lpwans standards and supported mobility, IEEE Commun. Surv. Tutor. 21 (2) (2018) 1561–1581, https://doi.org/10.1109/COMST.2018.2877382. Available DOI:.

[18] Brendan Normoyle, Jorge Luiz De Franco, Domnic White, Jonathan Roberts, Steve Wiseman, Bengt Johnnerfelt, Sentinel, fast detection of faulted transmission line arresters, in: 2020 IEEE/PES Transmission and Distribution Conference and Exposition (T&D), IEEE, 2020, pp. 1–5.

[19] https://www.bp.com/en/global/corporate/news-and-insights/reimagining-energy/drones-provide-bp-eyes-in-the-skies.html, accessed on January, 2020.

[20] T.v. Dam, K. Langendoen, An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks, SenSys'0, Los Angeles, California, USA, 2003, pp. 171–180, https://doi.org/10.1145/958491.958512. Available DOI:.

[21] P. Lin, C. Qiao, X. Wang, Medium access control with a dynamic duty cycle for sensor networks, in: WCNC 2004 /IEEE Communications Society, 2004, pp. 1534–1539, https://doi.org/10.1109/WCNC.2004.1311671. Available DOI:.

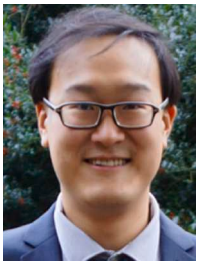
[22] Yanjun Sun, Shu Du, G. Omer, David Johnson, DW-MAC: a low latency, energy efficient demand-wakeup MAC protocol for wireless sensor networks, MobiHoc' 08 (2008) 53–62, https://doi.org/10.1145/1374618.1374627. Hong KongAvailable DOI:.

[23] L. Zhao, L. Guo, J. Zhang, H. Zhang, Game-theoretic medium access control protocol for wireless sensor networks, IET Commun. 3 (2009) 1274–1283, https://doi.org/10.1049/iet-com.2008.0383. Available DOI:.

[24] Woonsik Lee, Minh Nugyen, Arabinda Verma, Hwang Soo Lee, Schedule unifying algorithm extending network lifetime in S-MAC-based wireless sensor networks, IEEE Trans. Wireless Commun. 8 (2009) 4375–4379, https://doi.org/10.1109/TWC.2009.070370. Available DOI:.

[25] Wei Ye, Fabio Silva, John Heidemann, Ultra-low duty cycle MAC with scheduled channel polling, in: Proceedings of the 4th international conference on Embedded

networked sensor systems, ACM, 2006, https://doi.org/10.1145/1182807.1182839. Available DOI:.
[26] Michael Buettner, et al., X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks, in: Proceedings of the 4th international conference on Embedded networked sensor systems, ACM, 2006, https://doi.org/10.1145/1182807.1182838. Available DOI:.
[27] De-gan Zhang, Shan Zhou, Ya-meng Tang, A low duty cycle efficient MAC protocol based on self-adaption and predictive strategy, Mobile Netw. Appl. 23 (4) (2018) 828–839, https://doi.org/10.1007/s11036-017-0878-x. Available DOI:.
[28] Shu Du, Amit Kumar Saha, David B. Johnson, RMAC: A routing-enhanced duty-cycle MAC protocol for wireless sensor networks, in: IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications, IEEE, 2007, https://doi.org/10.1109/INFCOM.2007.174. Available DOI:.
[29] Hai-Ming Chen, Li Cui, Gang Zhou, A light-weight opportunistic forwarding protocol with optimized preamble length for low-duty-cycle wireless sensor networks, J. Comput. Sci. Tech. 32 (1) (2017) 168–180, https://doi.org/10.1007/s11390-017-1712-4. Available DOI:.
[30] Long Cheng, et al., Achieving efficient reliable flooding in low-duty-cycle wireless sensor networks, IEEE/ACM Trans. Netw. (TON) 24 (6) (2016) 3676–3689, https://doi.org/10.1109/TNET.2016.2549017. Available DOI:.
[31] L. Cheng, J. Niu, C. Luo, L. Shu, L. Kong, Z. Zhao, Y. Gu, Towards minimum-delay and energy-efficient flooding in low-duty-cycle wireless sensor networks, Comput. Netw. Chem. Lab., Symp. (134) (2018) 66–77, https://doi.org/10.1016/j.comnet.2018.01.012. Available DOI:.
[32] Azzedine Boukerche, Qiyue Wu, Peng Sun, Efficient green protocols for sustainable wireless sensor networks, IEEE Trans. Sustain. Comput. 5 (1) (2019) 61–80.
[33] Selahattin Kosunalp, Paul D. Mitchell, David Grace, Tim Clarke, Practical implementation issues of reinforcement learning based ALOHA for wireless sensor networks, in: ISWCS 2013; The Tenth International Symposium on Wireless Communication Systems, VDE, 2013, pp. 1–5.
[34] Dominic Charles White, Jonathan D. Roberts, Tautvydas Mickus, Paul D. Mitchell, David Grace and Yi Chu, "Communication network for monitoring a chain based network", U.S. patent US20200120010A1, 2020. Available: https://patents.google.com/patent/US20200120010A1/en?oq=US±2020%2f0120010±A1, accessed on January 2023.
[35] Product information available: https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1276, accessed on November, 2020.
[36] Product information available: http://www.ti.com/product/MSP430F2617#, accessed on November, 2020.
[37] Liya Zhao, Lihua Tang, Junrui Liang, Yaowen Yang, Synergy of wind energy harvesting and synchronized switch harvesting interface circuit, IEEE/ASME Trans. Mechatron. 22 (2) (2016) 1093–1103, https://doi.org/10.1109/TMECH.2016.2630732. Available DOI:.
[38] Seyedfakhreddin Nabavi, Lihong Zhang, Portable wind energy harvesters for low-power applications: a survey, Sensors (Basel, Switzerland) 16 (7) (2016) 1101, https://doi.org/10.3390/s16071101. Available DOI:.
[39] Aloÿs Augustin, Jiazi Yi, Thomas Clausen, William Mark Townsley, A study of LoRa: long range & low power networks for the internet of things, Sensors 16 (9) (2016) 1466, https://doi.org/10.3390/s16091466. Available DOI:.
[40] https://www.thethingsnetwork.org/article/ground-breaking-world-record-lorawan-packet-received-at-702-km-436-miles-distance, accessed on November, 2020.
[41] Z. Zhou, et al., Energy-efficient resource allocation for energy harvesting-based cognitive machine-to-machine communications, in: IEEE Transactions on Cognitive Communications and Networking, 5, 2019, pp. 595–607, https://doi.org/10.1109/TCCN.2019.2925025. Sept.Available DOI:.
[42] Yi Chu, Selahattin Kosunalp, Paul D. Mitchell, David Grace, Tim Clarke, Application of reinforcement learning to medium access control for wireless sensor networks, Eng. Appl. Artif. Intell. 46 (2015) 23–32, https://doi.org/10.1016/j.engappai.2015.08.004. Available DOI:.
[43] Haijun Liao, Zhenyu Zhou, Xiongwen Zhao, Lei Zhang, Shahid Mumtaz, Alireza Jolfaei, Syed Hassan Ahmed, Ali Kashif Bashir, Learning-based context-aware resource allocation for edge-computing-empowered industrial IoT, IEEE Internet Things J. 7 (5) (2019) 4260–4277, https://doi.org/10.1109/JIOT.2019.2963371. Available DOI:.
[44] Z. Zhou, Y. Guo, Y. He, X. Zhao, W.M. Bazzi, Access control and resource allocation for M2M communications in industrial automation, in: IEEE Transactions on Industrial Informatics 15, 2019, pp. 3093–3103, https://doi.org/10.1109/TII.2019.2903100. MayAvailable DOI:.
[45] R. Monica, L. Davoli, G. Ferrari, A wave-based request-response protocol for latency minimization in WSNs, IEEE Internet Things J. 6 (5) (2019) 7971–7979, https://doi.org/10.1109/JIOT.2019.2914578. Oct.Available DOI:.
[46] S. Chowdhury, A. Benslimane, C. Giri, Noncooperative gaming for energy-efficient congestion control in 6LoWPAN, IEEE Internet Things J 7 (6) (2020) 4777–4788, https://doi.org/10.1109/JIOT.2020.2969272. JuneAvailable DOI:.
[47] https://www.murata.com/eneu/products/connectivitymodule/lpwa/overview/lineup/type-abz-078, accessed on January 2023.
[48] https://www.rakwireless.com/en-us/products/lpwan-gateways-and-concentrators/rak2245-pihat, accessed on January 2023.



Yi Chu received his B.Sc. degree in electronic engineering from China Agriculture University, Beijing, China in 2008, the M.Sc. degree in communications engineering from the University of York in 2009, and the Ph.D. degree in electronic engineering from the University of York in 2014. He is now a Research Associate with the Centre for High Altitude Platform Applications (CHAPA), Department of Electronic Engineering, the University of York. His research interests include wireless and quantum communications applications on aerial platforms, communications system evaluation using software defined radio, multi-element antenna array, wireless signal propagation, physical layer network coding and intelligent medium access control for wireless sensor networks.

PAUL D. MITCHELL (Senior Member, IEEE) received the M. Eng. and Ph.D. degrees from the University of York, in 1999 and 2003, respectively. His Ph.D. research was on medium access control for satellite systems, which was supported by British Telecom. He has over 20 years research experience in wireless communications, and industrial experience gained at BT and DERA (now QinetiQ). He has been a member of academic staff with the Department of Electronic Engineering, University of York, since 2005, and is currently a Full Professor. Primary research interests lie in underwater acoustic communication networks, terrestrial wireless sensor networks, and communication protocols; including the development of novel medium access control and routing strategies. Other related interests include machine learning, traffic modeling, queuing theory, satellite, and mobile communication systems. He is an author of over 140 refereed journal and conference papers, and he has served on numerous international conference programme committees including ICC and VTC. He was a General Chair of the International Symposium on Wireless Communications Systems, in 2010. He currently serves as an Associate Editor of IET Wireless Sensor Systems journal, International Journal of Distributed Sensor Networks and MDPI Electronics, and has experience as a Guest Editor and as a reviewer for a number of IEEE, ACM and IET journals. He has secured more than >£2.3M + e4.7M funding as principal and coinvestigator. Current projects include Research Council grants on smart dust for large scale underwater wireless sensing, full-duplex underwater acoustic communications, as well as industrial projects. He is a member of the IET.

David Grace (S'95-A'99-M'00-SM'13) received his PhD from University of York in 1999, with the subject of his thesis being 'Distributed Dynamic Channel Assignment for the Wireless Environment'. Since 1994 he has been a member of the Department of Electronic Engineering at York, where he is now Professor (Research) and Head of Communication Technologies Research Group. He is also a CoDirector of the York - Zhejiang Lab on Cognitive Radio and Green Communications, and a Guest Professor at Zhejiang University. Current research interests include aerial platform based communications, cognitive green radio, particularly applying distributed artificial intelligence to resource and topology management to improve overall energy efficiency; 5G system architectures; dynamic spectrum access and interference management. He is currently a lead investigator on H2020 MCSA 5G-AURA and H2020 MCSA SPOTLIGHT. He was a one of the lead investigators on FP7 ABSOLUTE and focused on extending LTE-A for emergency/temporary events through application of cognitive techniques. He was technical lead on the 14-partner FP6 CAPANINA project that dealt with broadband communications from high altitude platforms. He is an author of over 220 papers, and author/editor of 2 books. He is the former chair of IEEE Technical Committee on Cognitive Networks for the period 2013/4. He is a founding member of the IEEE Technical Committee on Green Communications and Computing. In 2000, he jointly founded SkyLARC Technologies Ltd, and was one of its directors.

Jonathan Roberts has been the Principal Software Systems Engineer at IoT Lab, Corporate Technology TE Connectivity since 2014. His skills include developping embeded software and systems, digital signal processors and IoT applications.

Dominic White has been the Senior Software Systems Engineer at IoT Lab, Corporate Technology TE Connectivity since 2014. His skills include devlopping embeded software and systems, embded Linux applications and ARM programming.

Tautvydas Mickus received his Ph.D. in electronic engineering from the University of York in 2016. His research interests include Wireless Sensor Networks, Medium Access Control and Machine Learning. He was a lecturer in Klaipeda University at Faculty of Marine Technologies and Natural Sciences in 2017 and 2018. He is now the Mechanical Team Leader at Mars, Lithuania.