A clustered trail-based data dissemination protocol for improving the lifetime of duty cycle enabled wireless sensor networks

Richard Werner Pazzi[1] · Azzedine Boukerche[2] · Robson Eduardo De Grande[2] · Lynda Mokdad[3]

Abstract Wireless sensor networks (WSNs) usually consist of unmanned and self-organized sensor devices deployed on a target region for monitoring and target tracking purposes. Therefore, extending the lifetime of WSNs is critical to their proper operation. Static sink schemes have been studied extensively and several solutions were proposed to extend the lifetime of WSNs. Such static solutions are known for their bottleneck in the vicinity of a sink, where sensor nodes are more likely to be used as data forwarding points to a nearby sink. These nodes carry most of the data traffic and consequently deplete their energy resources faster than nodes deployed far off static sinks, which end up creating blind spots or even network partitions. A partitioned, i.e., disconnected network will cease to function properly in view of sink stations becoming unreachable. The consensus reached by the research community to solve this bottleneck, or at least to alleviate traffic and energy consumption near data sinks, is the use of mobile sinks. To this end, this paper presents a new data dissemination strategy (eTrail) that combines clustering, trail generation, and sleep scheduling techniques to extend network lifetime even further. Network lifetime is modeled and analyzed by means of a continuous time Markov chain. In addition, an extensive set of simulation experiments is presented and discussed. Results show that eTrail outperforms existing schemes in terms of network lifetime, while maintaining acceptable packet delivery reliability and latency.

Keywords Wireless sensor networks · Wireless networks · Mobility models · Performance evaluation · Network simulation

1 Introduction

Typical wireless sensor networks (WSNs) can be seen as a large number of miniaturized and battery-powered computer devices that house sensing, processing and communication modules. These very constrained sensor nodes can be programmed to self-organize into a wireless network in order to relay sensed data to a remote gateway or sink in a multi-hop fashion. Thus, WSNs can be used in a number of applications ranging from surveillance and environment monitoring to health care, military operations and even in homes [6]. Energy consumption in WSNs plays a major role since a number of deployment scenarios require that sensor nodes be left unattended for long periods of time. For instance, WSNs deployed over a target area for emergency response, e.g., catastrophes, natural disasters, terrorist attacks or border control, must operate non-stop for hours or even days. Even when deployed in a controlled

This work is partially sponsored by the Natural Sciences and Engineering Research Council of Canada (NSERC), Discovery Grant Program, and the NSERC DIVA Strategic Research Network.

✉ Richard Werner Pazzi
richard.pazzi@uoit.ca

Azzedine Boukerche
boukerch@site.uottawa.ca

Robson Eduardo De Grande
rdegr032@site.uottawa.ca

Lynda Mokdad
lynda.mokdad@univ-paris12.fr

1 DMCG Research Laboratory, University of Ontario Institute of Technology (UOIT), Oshawa, ON, Canada

2 PARADISE Research Laboratory - SITE, University of Ottawa, Ottawa, ON, Canada

3 Laboratoire LACL, Université Paris-Est, Créteil, France

environment such as in indoor monitoring, WSNs may be required to function properly for long periods of time during power shortages. Therefore, the tradeoff between WSNs lifetime and network performance has become one of the critical research issues to overcome.

A considerable number of energy-aware data dissemination solutions for WSNs can be found in the literature [4]. Most of the existing solutions employ static sinks. Such static approaches are known for their bottleneck in the vicinity of a sink, where sensor nodes are more likely to be used as data forwarding nodes to a nearby sink. These nodes carry most of the data traffic in WSNs and as a result deplete their energy resources faster than nodes deployed far off static sinks. This condition ends up creating blind spots and uneven communication loads, leading to network partitioning and reduced network lifetime. A partitioned, i.e., disconnected network will cease to function properly in view of sink stations becoming unreachable. Conversely, schemes that employ mobile sinks provide better load balancing and energy savings [7]. Methods that utilize mobile sinks like [8, 10, 25] are not affected by the bottleneck problem; mobile sinks are constantly changing their positions, thereby resulting in even energy consumption and traffic load within a sensor network. However, since sensor nodes are constrained devices with no global knowledge of the network, data must be disseminated through multi-hop paths. This leads to the problem of finding a path to a nearby mobile sink, which can induce communication overhead by frequent route updates. Khan et al. [14] conducted a thorough study of the impact of static versus mobile sinks on energy consumption of wireless sensor networks. The authors took into consideration the duty cycle of the nodes and their findings show that it takes more than simply deploying mobile sinks to achieve energy savings: it requires a careful planning of the duty cycle and mobility parameters to realize the benefits of mobile sinks. A thorough survey on algorithm design for communication in duty-cycled wireless sensor networks can be found in [13].

Previous works on mobile sink strategies assume either availability of global information of the network [1, 9, 26] or network-wide broadcasts [5, 18, 28] in order to discover paths to mobile sinks. Consequently, the potential energy advantage of using mobile sinks is overshadowed by extra energy consumption of frequent network broadcasts. Thus, how can a technique extend network lifetime while keeping frequent route updates local?

This paper describes the eTrail data dissemination mechanism: an attempt to answer the above question by a simple, yet efficient combination of clustering, duty cycle scheduling and trail formation. Basically, a mobile sink in eTrail works as a cluster-head by building clusters of sensor nodes periodically. As the mobile sink moves it sends beacons that act as if a trail is left along the sink's path. Sensor nodes that receive these beacons will coordinate and update their routing information in order to maintain an up-to-date path to the mobile sink. This trail-based approach guarantees that any node in the cluster or along the trail has a route to the mobile sink. Besides, this mechanism is restricted to local broadcasts and induces minimal communication overhead. An overview of eTrail is depicted in Fig. 1. eTrail also features a sleep scheduling scheme for duty cycle enabled sensor nodes. The key concept of the sleep scheduling scheme is to let every single sensor node in a cluster sleep, unless one of its children nodes tells it not to. Since the radio module is responsible for most of the energy consumption of a sensor node, putting a node to sleep will basically switch its radio off, thereby conserving energy and prolonging network lifetime. The key challenge lies in how to schedule nodes to sleep while keeping valid paths of awake nodes to guarantee data dissemination to mobile sinks. In addition, a local path recovery mechanism is triggered whenever a path is disconnected.

This paper also discusses the modeling and analysis of network lifetime by means of a continuous time Markov chain. Thereafter, a set of simulation experiments is presented and results reveal that eTrail improves on network lifetime significantly while maintaining reasonable packet delivery reliability and latency compared to selected previous works.

2 Related works

A number of mobile data gathering protocols has been proposed in the literature in order to solve or minimize the problems induced by static sink solutions. These protocols target energy efficiency by adopting strategies to constrain the broadcasting of mobile sink's position information to a limited area, while maintaining network connectivity. Network connectivity and network coverage fall under the umbrella of topology control, which is critical for the efficient operation of WSNs. Please refer to [15] for a comprehensive survey on topology control and guidelines and best practices for WSNs. Sink mobility patterns also play an important role in the design of efficient data dissemination protocols. Sinks can move according to three different patterns: predetermined, controlled, and random mobility. Protocols designed for predefined and controlled sink mobility exploit predefined paths and techniques to control sink mobility in order to cover areas with lower node density, maintain connectivity, and balance energy consumption [2, 9, 17, 19, 22, 31]. Conversely, protocols that assume random sink mobility must keep sensor nodes aware of the sink position. Random sink mobility broadens

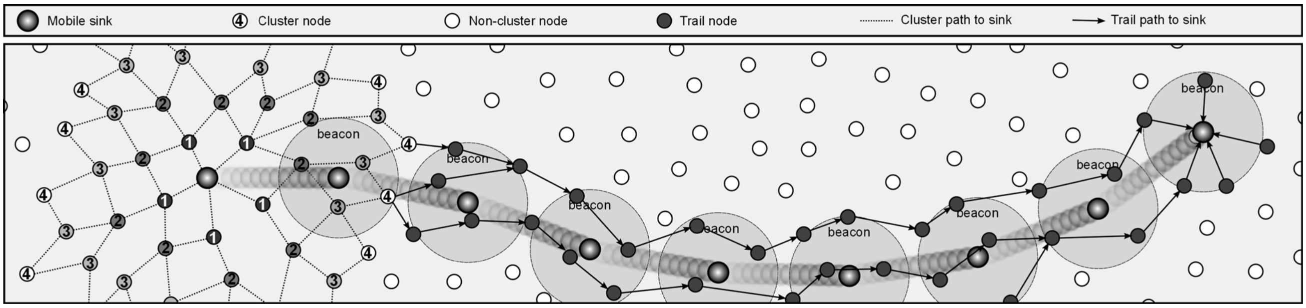

Fig. 1 Overview of eTrail: a mobile sink builds a cluster and broadcasts beacon messages as it traverses the sensor field. Sensor nodes that receive a beacon will update their routing information and the process will end up creating a trail to the mobile sink until a new cluster is built and the process is restarted. Any node in the cluster or on the trail will have a valid path to relay data to the sink

the suitability of an approach to a diverse range of applications. Moreover, controlled sink mobility may fail when sensor nodes are deployed in areas of difficult access.

According to a survey by Kai Han et. al. [13], there are four different categories of duty-cycle algorithms for a WSN: (1) synchronized versus non-synchronized; (2) generalized versus simplified; (3) static versus dynamic, and; (4) reliable versus unreliable links. Since nodes in eTrail work in rounds or cycles, nodes that can enter sleep mode will do so for the duration of a round and must be awake for the next cluster configuration phase. It is in this phase that sleep/awake modes are decided for the next round. Therefore, eTrail adopts a dynamic non-synchronized approach where the duration of a dormant state is given by the mobile sink during the cluster configuration phase: the "cluster age" value carried by a CLUSTER_CFG message. This way eTrail guarantees that a node in sleep mode will be active by the end of the current round.

Wu et al. proposed the Dual Sink protocol [32], which uses both static and mobile sinks. Basically, when the mobile sink moves to a new site along a predetermined square-shape path, it only broadcasts beacons to a nearby subset of nodes in the network. The static sink broadcasts a HELLO message to all nodes in the network only once at the initialization phase. In the end, each node in the network will have at least one route to the static sink. Nodes that have routes to both static and mobile sinks can send data to the closest sink. Dual-Sink is one of the first protocols that addresses the problems caused by repeated network-wide broadcasting. However, the drawback of Dual-Sink is its scalability: the mobile sink only broadcasts beacons to nearby sensor nodes that are less than two or three hops away, which might not be enough to reach other areas of a wireless sensor network. Consequently, Dual-Sink may behave as a simple static sink approach in large-scale sensor networks.

The Efficient Routing protocol proposed by Fodor et al. [8] supports multiple mobile sinks with no restrictions on sink mobility patterns. It uses restricted broadcasting to update the location of mobile sinks in the network. Basically, each node maintains a routing table containing a list of neighbor nodes that can be used as relays to forward data to a mobile sink. When a node receives a DATA message, it selects the nearest mobile sink with the smallest registered cost (which is a concept similar to the hop distance in Dual-Sink) as its destination sink and it uses the corresponding intermediate node as its next hop. Efficient Routing employs a mechanism slightly different from Dual-Sink to process sink location updates. The concept of cost can be chosen arbitrarily such as the number of hops from the sink or some QoS related metrics. Two types of costs are used in this protocol: the registered cost of each sensor node to a specific mobile sink, and the actual cost propagated in sink location update messages. A threshold is then defined relative to the ratio of the registered cost and the actual cost. Upon the reception of a sink location update message, the sensor node compares the cost in the message with its own registered cost. If the difference between the two costs exceeds a certain predefined threshold, the routing information, as well as the registered cost of this sensor, are updated and the message is further forwarded to other sensors. Conversely, in case the difference does not exceed the threshold, the update message is simply discarded. The threshold must be carefully set and tested as its value may affect significantly the performance of the protocol.

The Local Update-based Routing Protocol (LURP) proposed by Wang et al. [29] suggests a new mechanism to constrain the location broadcasting within a local area rather than among the entire sensor network. In LURP, as long as the mobile sink moves within a small circular area called the "destination area", it broadcasts location updates only to the sensor nodes inside the destination area. When the sink moves out of the local area, a new destination area is built on the fly through one round of network—wide broadcasting and then the same local update process is repeated within the new destination area. LURP is a flat data dissemination protocol. However, unlike other flat protocols discussed previously in this thesis, LURP assumes sensor nodes in the network to be location-aware, i.e.,each node knows its own location, its one-hop neighbors and the location of the sink by certain localization algorithms.

The Adaptive Location Update-based Routing Protocol (ALURP) [30] is an improved version of the LURP protocol. In ALURP, the mobile sink adaptively adjusts its local broadcasting range as it moves within the destination area. This improvement further decreases the overhead of sink location updates and reduces average energy consumption per sensor node.

The Line-based data dissemination protocol (LBDD) [12] is also a virtual infrastructure based protocol for WSNs with mobile sinks. LBDD uses a virtual vertical line or strip area constructed in the central field of the wireless sensor network that divides the network into two equal parts. Sensor nodes within the boundaries of this virtual area are called the inline nodes. The line or strip area acts as a rendezvous area for sensor data and sink queries. Upon the detection of an event, the source transmits the gathered data towards the virtual area in the center of the network. The data is stored on the first inline node encountered on the forwarding path from the source to the rendezvous area. When the mobile sink requests event data, it also sends a query towards the rendezvous area. Once inside the rendezvous area, the sink query travels linearly until it encounters the data storing inline node. Then, based on the location information of the mobile sink propagated in the sink query, event data is geographically routed to the sink.

The Railroad protocol proposed by Shin et al. [24] adopts an approach similar to the LBDD protocol. It constructs a virtual square-shape strip area that is called the rail in the central area of the sensor network. Sensor nodes located inside this strip area become the rail nodes. When the source detects an event, it stores the event data locally and sends the corresponding meta-data (e.g., a data announcement message) towards the nearest rail node. The mobile sink sends its data queries to the rail as well. Once inside the rail, the query travels around the square-shape strip area until it reaches a rail node that holds the meta-data about the event and the related source node. Event data is then directed from the source to the mobile sink geographically based on the source location information provided in the meta-data and the sink location information provided in the sink query. The common drawback of the Railroad protocol and the LBDD protocol (and other similar virtual infrastructure based protocols) is that they both use a fixed virtual area in the sensor network to assist the data dissemination process. In homogeneous WSNs where all deployed sensors have the same energy resources, protocols like Railroad and LBDD are usually not applicable due to the hot spot problem caused by the fixed rendezvous area in the sensor field. A summary of the protocols discussed in this session is given in Table 1. As can be seen in Table 1, eTrail, Dual-Sink and Efficient Routing belong to a category of protocols that do not require location awareness, i.e., these protocols do not use position information to route data to a sink. This is the reason why eTrail is compared against protocols in this category when evaluating its performance.

Gu et al. [11] proposed an interesting approach to reduce data latency in WSNs that employs mobile aerial vehicles (or drones) to collect data from the sensor field. These drones are equipped with a sensor and follow the proposed algorithm to fly over the sensor field and reach specific regions. They also deploy a central station in the sensor field that is responsible for coordinating and dispatching drones and providing a battery recharge station, which eliminates the energy hole problem for the mobile sinks. Simulation experiments show that their proposed solution can minimize data latency in WSNs with mobile sinks. The only drawback is the bottleneck of the central station used to dispatch and recharge drones.

The authors in [3] have recently proposed a new prediction-based target tracking strategy for WSN with mobile sinks, called t-Tracking, that uses “face prediction” instead of the traditional approach of predicting the position of a target. Simulation results showed improved localization accuracy and energy efficiency. Although not directly related to the proposed scheme in this paper, their ideas and algorithms could be slightly tweaked to aid nodes find a mobile sink, which would act as a target or a “rendezvous area” where packets would be relayed to and intercepted by a mobile sink.

The Energy-efficient Delay-Aware Lifetime-balancing (EDAL) routing protocol [33] uses several metrics to select the best data paths in a WSN taking into account packet delay and load balancing requirements. EDAL extends network lifetime by using links with more remaining power, where links are assigned “weigths” based on the battery level of nodes on that path. Simulation results show that EDAL improves network lifetime while having little impact on packet delay. EDAL does not consider mobile sinks, and its sleep-awake duty cycle scheme is based on node redundancy and density. The proposed eTrail also uses the battery level in its routing decisions like EDAL. However, instead of assigning weights to links, each node in eTrail decides if its parents nodes will sleep or be awake for a round based on their energy level values, which are disseminated (piggybacked) during the cluster formation phase in order to minimize traffic. A direct performance comparison with EDAL is left as future work.

Another approach that one could take on the design of data communication protocols from sensor nodes to mobile

Table 1 Summary of WSNs protocols with mobile sinks

Protocol	Event reporting	Disseminated info	Sink mobility	Virtual structure	Location awareness
eTrail	Continuous	Sink location (beacons) and data	Random	None	No
Dual-Sink [32]	Continuous	Sink location and data	Predefined	None	No
Efficient Routing [8]	Continuous	Sink location and data	Random	None	No
LURP [29]	Continuous	Sink location and data	Random	None	Yes
ALURP [30]	Continuous	Sink location and data	Random	None	Yes
LBDD [12]	Event/query	Sink location and data	Random	Strip	Yes
Railroad [24]	Event/query	Sink location, Meta-data and data	Random	Railroad	Yes

sinks is through Delay Tolerant Networks (DTNs). For instance, Yuanyuan Zeng et al. [34] proposed a directional routing and scheduling scheme (DRSS) that can be easily adapted to work with the proposed eTrail. Although eTrail considers static nodes, these could hold or relay data to a "rendezvous area" until a mobile sink is nearby to collect data. The only drawback is the increased packet delay, which would render eTrail suitable to delay-tolerant applications only.

Usually, algorithms for coverage and lifetime maximization deploy a large number of sensors in an area (high density) and then schedule redundant nodes to a dormant state. Eventually the active nodes will switch roles with dormant ones, and thus network lifetime is improved. Soumyadip Sengupta et al. [23] took a different approach where the algorithm is triggered in the event of a node failure due to battery depletion or mechanical/electrical problems. Their scheme first deploys a dense network and the active nodes are selected. When a node fails, the algorithm kicks in to reschedule the nodes assuming there are available nodes and network connectivity. eTrail uses a simplified algorithm where, in the event of a node failure, a child node will trigger a path reconstruction mechanism and re-establish communication along a path. The scheme proposed in [23] could be adapted to eTrail in order to manage node failures in the presence of dormant nodes.

The Density-based proactivE data dissEmination Protocol (DEEP) [27], which combines a probabilistic flooding with a probabilistic storing scheme employs uncontrolled mobile sink trajectory to collect data in WSNs. Although both the proposed scheme and DEEP share a few similarities, the purpose of these schemes are quite different. Main difference is that the proposed scheme is suitable for real-time data collection, whereas DEEP adds significant packet delay and it is, therefore, more suitable for delay-tolerant applications. In eTrail, a mobile sink starts receiving data right after a cluster is formed and it continues receiving data through a trail. DEEP, on the other hand, disseminates data to certain areas in the network. Nodes in those areas will store and hold data until a mobile sink is nearby, which then collects the stored data (this process adds delivery delay). The proposed scheme integrates clustering and duty-cycle modes (sleep/awake) to save energy even further and extend network lifetime. The decision on which nodes sleep and which ones stay active is completely local and piggybacked during the cluster formation phase.

3 Protocol design

3.1 Assumptions

This paper considers a wireless sensor network composed of s static nodes and m mobile sinks deployed over an area of $n \times n$ m. Sensor nodes relay data to mobile sinks by multi-hop paths. Sensor nodes considered in this paper use symmetric links. In addition, sensor nodes have limited energy, communication and storage resources. This paper assumes that mobile sinks have unlimited energy they are considered the final destination of sensor data. Sensor nodes do not need to know their positions; the proposed scheme does not rely on node's positions. However, mobile sinks may rely on Global Positioning System (GPS) or indoor positioning systems in order to minimize the number of beacons. If we consider vehicles as mobile sinks, the GPS assumption is realistic since it is cheap and easy to equip cars with GPS receivers. This assumption can be relaxed if one considers fixed or speed-based approaches to determine beacon intervals. In this work we consider the distance traveled by a mobile sink as the criterium to decide when to send beacons.

3.2 The eTrail protocol

eTrail is an attempt to extend network lifetime by solving the bottleneck problem of static sink solutions while minimizing routing updates induced by mobile sinks. Basically, eTrail works in rounds and a mobile sink builds a tree (cluster) in every round. Sensor nodes that belong to a cluster will relay data its mobile sink through multi-hop links. eTrail includes a trail-based approach similar to breadcrumb solutions for mobile networks. After creating a cluster, a mobile sink will leave a "trail" by broadcasting beacon messages to nearby sensor nodes, which will then update their routing information. eTrail includes a duty-cycle mechanism that switches clustered nodes to sleep mode, thereby saving energy resources. A failure recovery mechanism is also devised in order to avoid malfunctioning nodes to disrupt data dissemination through the network.

3.2.1 Cluster/tree configuration

Each mobile sink in the network will create a cluster per round. At the end of a round (or cluster lifetime), mobile sinks start new clusters (given that sinks have moved). eTrail uses the idea of restricted flooding with a specific time-to-live (*ttl*) to build clusters/trees rooted at mobile sinks as proposed in [21]. Basically, a mobile sink starts a cluster by broadcasting a CLUSTER_CFG message. This message carries the sink ID, the sender's address, the hop count (*hop*), the cluster age, and the sender's remaining energy, as depicted in Fig. 2. Upon receiving a CLUSTER_CFG message, a node will first check the packet's hop count *hop* for the specific sink ID. If *hop* is less than the node's hop count (i.e., received from previous CLUSTER_CFG messages), this node will set the sender node as its next hop, increment *hop* and forward the CLUSTER_CFG message if *ttl* has not reached 0. Conversely, if *hop* is greater or equal to the node's hop count the node will simply discard the packet. Each node also will store the remaining energy of its parent nodes (i.e., neighbour nodes one hop level down the path) to identify the parent nodes that will stay "awake" during the current round. A detailed description of the cluster configuration mechanism is given in Algorithm 1. At the end of this process each node in a cluster will have a path to a mobile sink. An eTrail round will eventually come to an end and a new cycle will begin by mobile sinks sending CLUSTER_CFG messages. Although the proposed strategy requires that nodes have clocks, they do not need to be synchronized; all they need is a timer. The message CLUSTER_CFG, which initiates the tree/cluster configuration, carries a field called "cluster age". The "age" of a cluster defines the length of a round: routing tables' settings are reset and the mechanism restarts. The age of a cluster is used to limit the length of a trail. Long trails can induce high packet delays and losses.

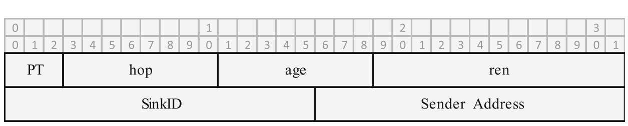

Fig. 2 Cluster_CFG packet header. PT: packet type; hop: number of hops; age: cluster's age; ren: sender's remaining energy; SinkID: sink identifier; Sender address

Algorithm 1 CLUSTER CONFIGURATION

```
CLUSTER_CFG Packet: pkt
sinkID; {ID or address of the sink that generated CLUSTER_CFG}
sID; {Sender's ID or address}
sREN; {Sender's remaining energy}
hop; {Number of hops the pkt has traveled}
ttl; {Packet's Time-to-Live}
age; {Lifetime of the cluster}

Local variables
chop; {The node's current hop level}
maxREN = 0.0; {Stores the highest sREN received from neighbor nodes}
Tn; {Timer to send NO_SLEEP message}
Ts; {Timer to go to sleep}

Routing Table: RTable
sinkID; {Sink ID or address}
dest; {Destination ID or address (relay node)}

Action: {Node received a CLUSTER_CFG packet}
SetTimer(Tn); {Timer to collect pkts from neighbors}
  {Will process pkts received from lower hop-level nodes only}
if (pkt.hop < chop) then
    chop ← pkt.hop;
    {Node with highest energy is set as destination}
    if (maxREN < pkt.sREN) then
        maxREN ← pkt.sREN;
        RTable.sinkID ← sinkID;
        RTable.dest ← sID;
    end if
else
    Drop(pkt); {Silently drops pkt.}
    EXIT;
end if

SetTimer(Ts); {Node will go to sleep in Ts seconds}
if (ttl > 0) then
    pkt.hop ← pkt.hop + 1;
    pkt.ttl ← pkt.ttl - 1;
    Forward(pkt); {Packet pkt is forwarded}
else
    EXIT;
end if

Action: {Timer Tn goes off}
  {Node will send a NO_SLEEP message to node address RTable.dest}
Send(NO_SLEEP, RTable.dest);

Action: {Node received a NO_SLEEP message}
  {Node is requested to be awake during this round}
CancelTimer(Ts);

Action: {Timer Ts goes off}
  {Node will enter sleep mode for the duration of the current round}
Sleep(age);
```

3.2.2 Duty-cycle mechanism

As mentioned in the cluster configuration phase, child nodes will gather the energy information of their parent nodes in order to identify the nodes that will stay awake for the duration of the current round. The central idea is to keep only one parent node awake for each child node during a round. The awake parent node will be used as data relay towards the sink. During the cluster configuration phase, a mobile sink broadcasts a CLUSTER_CFG, which is then forwarded by sensor nodes. Nodes forward this message along with their remaining energy levels. Child nodes will receive CLUSTER_CFG during a specific time interval T_n from all parents and identify the parent with the highest energy level. When T_n expires, the child node sends a NO_SLEEP message to the selected parent node. This message informs the parent to be awake for the duration of the round. After a sensor node forwards CLUSTER_CFG it will set a timer T_s to go to sleep. This timer will be canceled only if the node receives a NO_SLEEP message from a child. This mechanism is described in Algorithm 1. Energy consumption will be balanced after a few rounds: nodes that sleep in a round will eventually be chosen to be awake in future rounds since they will probably have higher energy levels due to savings in sleep mode. The sleeping period should not exceed the round duration, which is specified by the field *age*. Thus, the sleeping nodes will be awake when the next cluster creation round begins. The sleep scheduling scheme guarantees that sensor nodes in WSNs have at least one neighbor node awake in order to relay data to a mobile sink, i.e., connectivity coverage. This paper also assumes that sensor nodes are capable of waking their radios up when an event is detected by their sensor modules. This assumption is valid since a number of hardware solutions currently implement such duty-cycle feature like the iMote2 IPR2400 processing and sensing module [20].

In order to clarify the sleep-awake process, consider the wireless sensor network depicted in Fig. 3(a), where a mobile sink has created a cluster with size 4 ($ttl = 4$). The numbers inside the circles (nodes) represent hop levels. Basically, during the cluster configuration phase, the sink broadcasts a CLUSTER_CFG message. Nodes 1A, 1B and 1C in Fig. 3(b), for instance, receive that message, process it according to algorithm 1, and forward it to their neighbors along with their energy levels. Node 2D receives CLUSTER_CFG from nodes 1B and 1C. In this case, node 2D selects its active parent for the current round to be the node with higher energy level (node 1B) by sending a NO_SLEEP message, which tells node 1B to remain awake for the duration of the round. This process continues until CLUSTER_CFG reaches the last nodes ($ttl = 4$). Nodes that do not receive a NO_SLEEP message will go to sleep during the current round, e.g., nodes 2B, 2D, 3C, etc., as shown in Fig. 3(c).

3.2.3 Trail generation

Right after the cluster configuration, all nodes will have a route to send data to the sink. However, since the sink is moving, its one-hop neighbors will eventually be out of range. As discussed earlier, a simple solution would be to make the sink send CLUSTER_CFG messages more frequently, such that nodes would have up-to-date routes to the sink. Depending on the size of the clusters (we can have the entire network as a single cluster, for instance), the potential energy advantage of using mobile sinks would be overshadowed by extra energy consumption of frequent clustering. Instead, eTrail is based on the concept of local, limited area route updates to achieve connectivity with mobile sinks. To this end, a mobile sink in eTrail sends BEACON messages to a limited number of nodes as it

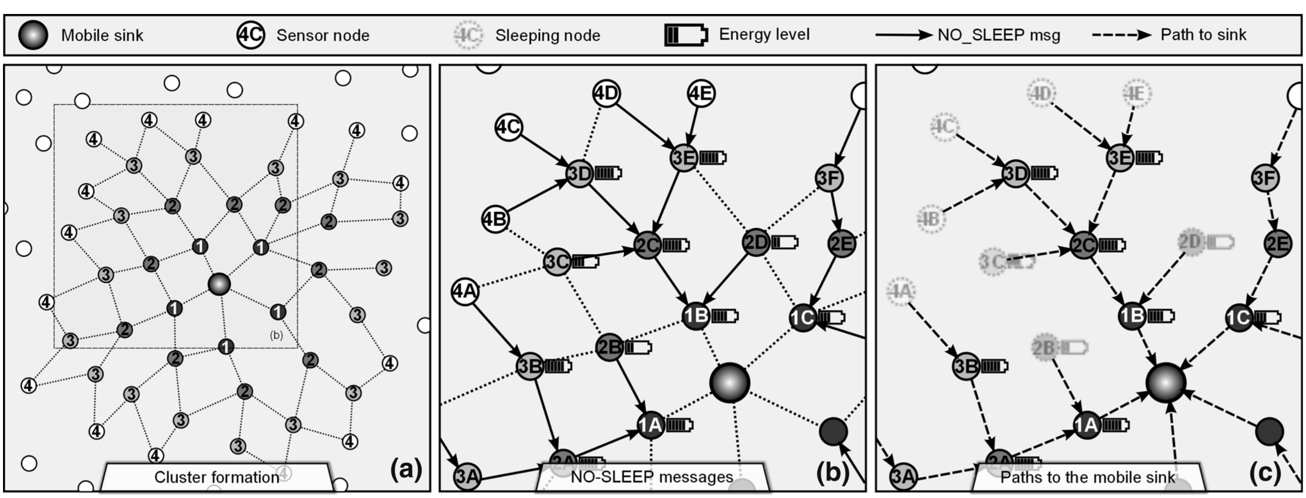

Fig. 3 **a** A mobile sink starts cluster configuration (ttl = 4). Nodes forward the configuration message along with their remaining energy levels; **b** nodes that receive a configuration message will select their parent node as the one with more energy level by sending the parent a NO_SLEEP message; **c** nodes that did not receive a NO_SLEEP message will turn their radios off (sleep) during the current round (e.g., nodes 2B, 2D, 3C, 4A, 4B, 4C, 4D and 4E)

moves. By sending BEACON messages with $ttl = 2$, a mobile sink creates a trail of nodes that is used to deliver data to the mobile sink. The mobile sink uses its traveled distance since the last beacon to decide the best time to send another BEACON message.

Basically, nodes that receive a BEACON message update their routing information based on hop level similarly to the cluster creation phase. The trail generation mechanism is depicted in Fig. 1. Beacon rate should be properly set in order to accommodate the sink's movements. If this rate is too low, the sink could move far away from the last point of attachment and the trail would break. Thus, the beacon rate is set dynamically depending on the traveled distance, i.e., the sink displacement D_s since the last beacon. In this paper, the displacement threshold is set to be equal to the communication range of a sensor node, or $D_s = r$. Thus, when the sink moves away from the position at which the last beacon was sent, a BEACON message is sent when this displacement reaches D_s. As can be seen in Fig. 4, by setting $D_s = r$ we can guarantee that sensor nodes within a distance *dist* from the mobile sink will receive at least one beacon. This way the algorithm has a better chance to form complete trails, i.e., no trail breaks. Considering the triangle in Fig. 4, *dist* can be calculated by Pythagora's Theorem.

Thus, considering a communication range r of 100 meters, the nodes that are within a distance *dist* of approximately 86.6 meters from the sink's path will receive at least one beacon as the sinks passes by. Furthermore, a trail has the same age as the cluster. Nodes in a trail will reset their routing information when the trail/cluster age expires.

3.2.4 Failure recovery scheme

Path breaks in WSNs will eventually happen due to a variety of reasons: (1) a node in the path runs out of energy; (2) a malfunctioning node; (3) a node shifts to sleep mode when it is supposed to be awake, *e.g.*, a NO_SLEEP message from a child node does not reach the target parent node due to packet collision. To this end, eTrail features an overhearing-based data dissemination failure recovery mechanism. Basically, suppose that a node A relays a data packet to its next neighbor (node B). Node A triggers the failure recovery mechanism if it does not overhear the same data packet being forwarded by node B. Thus, node A broadcasts a HELP message to its immediate neighbors. This message carries the hop count *hop* of node A. A neighbor node will reply to node A with a HELP_REP message only if it has a route to a mobile sink and its *hop* count is less than the hop count of node A. This is necessary to avoid loops and to minimize the number of replies. Node A waits for replies for a specific timeout and it chooses the node with lowest hop count to the sink. Thus, node A adds this neighbor as its new route and a path is reestablished. This mechanism is used either by cluster member nodes and trail nodes.

3.3 Network lifetime modeling and analysis

This section presents the modeling and analysis of network lifetime of eTrail by means of a continuous time Markov chain. We consider n sensors in the network, where

- i is the total number of sensor nodes in "NO-SLEEP step";
- j is the total number of sensor nodes in "SLEEP step".

We assume that the duration of "SLEEP step" follows an exponential distribution with rate μ and the duration of NO-SLEEP step follows an exponential distribution with rate λ. We also assume that lifetime follows an exponential distribution with rate γ.

The model can be illustrated by a continuous time Markov chain denoted by $X(t)$. To describe this chain, we

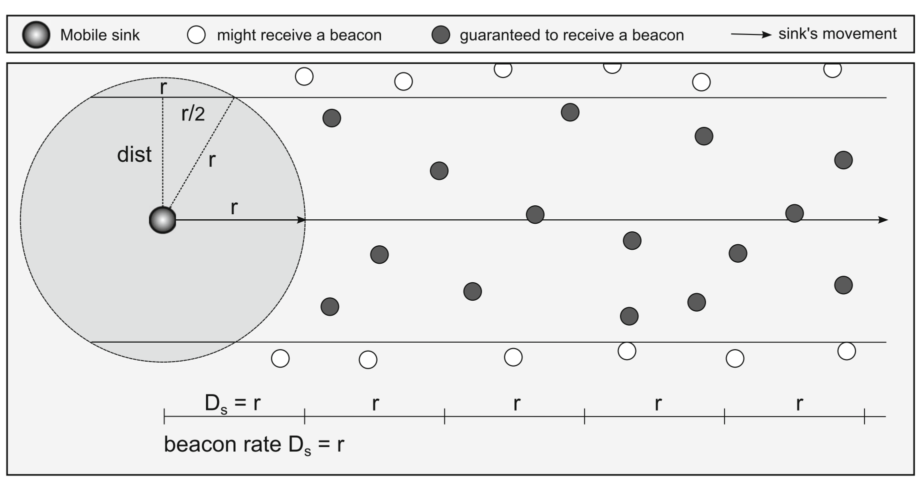

Fig. 4 Beacon rate is set based on the sink displacement $D_s = r$. The *dark* nodes will receive at least one beacon as the mobile sink passes by

define the state x by (i,j), where $\forall i, 0 \le i \le n$ and $\forall j, 0 \le j \le n$.

We denote its stationary probability distribution by $\pi(x)$. The model can be described by a multi-dimensional Markov chain as given in Fig. 5 (for $n = 4$).

The size of the Markov chain is given by:

$$\sum_{j=0}^{n}(n-j+1) = \frac{n(n+3)}{2} + 1$$

Since we are interested in modeling network lifetime, we need to understand the behavior of $X(t)$, by discussing its state evolution and transition rates. Thus, from a state $x = (i,j)$, and according to the different events that happen, we have the following transitions:

$$\begin{aligned} x &\rightarrow (x_i - 1, x_j + 1), withrate x_i \lambda IF(x_i \ge 1) AND (x_i + x_j \le n) \\ &\rightarrow (x_i + 1, x_j - 1), withrate x_j \mu IF(x_j \ge 1) AND (x_i + x_j \le n) \\ &\rightarrow (x_i - 1, x_j), withrate x_i \gamma IF(x_i \ge 1) \\ &\rightarrow (x_i, x_j - 1), withrate x_j \gamma IF(x_j \ge 1) \end{aligned}$$

Network lifetime corresponds to the time to reach state $s_{0,0}$ by considering the initial state $s_{n,0}$. Thus, the time to reach state $s_{0,0}$ from a state $s_{i,j}$ is given by:

$$T(s_{i,j}) = Soj(s_{i,j}) + \sum_{s_{i,j} \rightarrow s_{i',j'}} P[s_{i,j}, s_{i',j'}] T(s_{i',j'})$$

where $Soj(s_{i,j})$ is the sojourn time in state $s_{i,j}$. $P[s_{i,j}, s_{i',j'}]$ is the transition probability from state $s_{i,j}$ to $s_{i',j'}$.

For example, consider the sojourn time of state $s_{1,1}$:

$$\begin{aligned} T(s_{1,1}) = &\frac{1}{\lambda + \mu + \gamma + \gamma} + \frac{\lambda}{\lambda + \mu + \gamma + \gamma} T(s_{0,2}) \\ &+ \frac{\mu}{\lambda + \mu + \gamma + \gamma} T(s_{2,0}) \\ &+ \frac{\gamma}{\lambda + \mu + \gamma + \gamma} T(s_{0,1}) + \frac{\gamma}{\lambda + \mu + \gamma + \gamma} T(s_{1,0}) \end{aligned}$$

The calculation of sojourn time, considering the different values that i and j can take, is the following:

- Case 1: For any state $s_{i,j}$ such that $i \neq 0$ and $j \neq 0$, the sojourn time is given by:

$$\begin{aligned} T(s_{i,j}) = &\frac{1}{i\lambda + j\mu + i\gamma + j\gamma} + \frac{i\lambda}{i\lambda + j\mu + i\gamma + j\gamma} T(s_{i-1,j+1}) \\ &+ \frac{j\mu}{i\lambda + j\mu + i\gamma + j\gamma} T(s_{i+1,j-1}) \\ &+ \frac{i\gamma}{i\lambda + j\mu + i\gamma + j\gamma} T(s_{i-1,j}) \\ &+ \frac{j\gamma}{i\lambda + j\mu + i\gamma + j\gamma} T(s_{i,j-1}) \end{aligned}$$

- Case 2: For any state $s_{i,j}$ such that $i = 0$ and $j \neq 0$, the sojourn time is given by:

$$T(s_{0,j}) = \frac{1}{j\mu + j\gamma} + \frac{j\mu}{j\mu + j\gamma} T(s_{1,j-1}) + \frac{j\gamma}{j\mu + j\gamma} T(s_{0,j-1})$$

- Case 3: For any state $s_{i,j}$ such that $i \neq 0$ and $j = 0$, the sojourn time is given by:

$$T(s_{i,0}) = \frac{1}{i\lambda + i\gamma} + \frac{i\lambda}{i\lambda + i\gamma} T(s_{i-1,1}) + \frac{i\gamma}{i\lambda + i\gamma} T(s_{i-1,0})$$

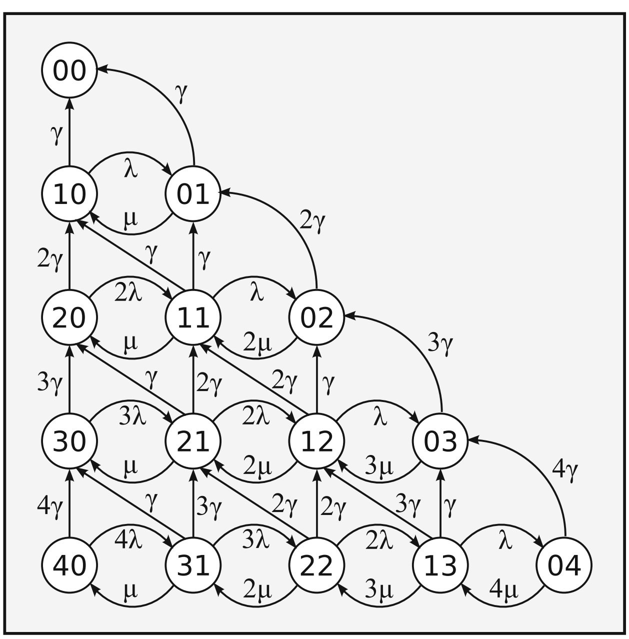

Fig. 5 Markov chain for $n = 4$

4 Performance evaluation

Simulation experiments were conducted to evaluate the performance of eTrail and compare it with existing data dissemination protocols with mobile sinks (*e.g.*, Dual-Sink [32] and Efficient Routing [8], and a modified version of eTrail in which the sleeping mechanism is turned off in order to assess the performance of this mechanism in isolation. The protocols were implemented in the network simulator ns-2. All protocols are evaluated under the same conditions and scenarios. In all simulations, sensor nodes are deployed on a square area following a disturbed grid topology, in which the nodes' coordinates are displaced by a limited random distance within a grid point in order to avoid large differences in node density (blind spots). In addition, data source nodes are randomly selected from the pool of nodes and the initial positions of mobile sinks are also randomly generated in each simulation execution. Except for the lifetime evaluation experiments, sensor nodes are given enough energy to complete an entire simulation execution. Mobile sinks move according to the random way-point mobility model. Simulation parameters are based on the iMote2 specification [20] and are summarized in Table 2. Average results are calculated from 10 executions with a confidence interval of 95 %.

In addition, there are protocol-specific parameters for the selected schemes that were taken into consideration in the simulations for the sake of fairness. These parameters are not clearly defined in their original publications. Therefore, they were configured according to their best performance through simulation: a mobile sink in Dual-Sink sends out HELLO messages every 20 s; the *TTL* of the mobile sink's HELLO messages is set to 5; and the cost change threshold in Efficient Routing is 60 %.

4.1 Power consumption

The average power consumption of eTrail and the selected protocols is evaluated in this first set of experiments. As observed in Fig. 6(a), the number of mobile sinks has little to no impact on average power consumption. The proposed eTrail protocol shows significantly better energy savings compared to the selected approaches. The main reason is the eTrail's duty-cycle mechanism, in which a number of sensor nodes enter sleep mode during a round. This performance gain is evidenced by the results of eTrail without sleeping, which employs the same trail-based routing update algorithm as eTrail, but it does not utilize duty-cycle sensor nodes. Thus, eTrail without the sleeping mechanism performs similarly to the selected approaches.

Table 2 Default simulation parameters

Parameter	Value
Radio range	100 m
TX power	0.033 W
RX power	0.033 W
Sleep power	0.003 W
Switch mode power	0.0001 W
Max sink speed	20 m/s
Data rate	1 pkt/s
Round period (age)	50 s
BEACON rate	Every 100 m
Bandwidth	38.4 Kbps
Packet size	44 bytes
Number of nodes	1024
Number of sinks	2–12
Number of sources	4–10
Deployment area	2000 × 2000 m

In eTrail, sensor nodes switch to sleep mode if they do not receive a NO_SLEEP message. When in sleep mode, a node's radio is turned off. When the mobile sink broadcasts trail beacons on the sensor field, much fewer nodes receive and forward trail beacons, resulting in better energy utilization.

The protocols were also evaluated under different node densities (number of nodes by square meter). As can be observed in Fig. 6(b), node density has an impact on eTrail's energy consumption per node. As expected, eTrail is more energy efficient in denser areas. This is due to the number of neighbors a node has. Since a node chooses only one neighbor as relay node per round, the probability that the other neighbors will sleep is very high. In a low density scenario, less nodes will sleep since they will be needed by their neighbors in order to relay data to the sink. It can also be observed in Fig. 6(b) that node density has no effect on energy consumption of the other protocols. The reason for this is that nodes do not use a duty-cycle mechanism and waste most of their energy in idle state.

In Fig. 6(c), energy consumption is assessed based on the number of nodes in the network (network size). Note that the deployment area varies as to maintain a constant node density. As can be observed, network size has no significant impact on energy consumption in all the selected protocols. All the protocols scale well with the network size since energy consumption per node does not depend on network size, i.e., although data packets travel through longer paths, the energy used by each node to send the packet is the same.

In addition, we observed that sensor nodes spend more time in idle state than in any other state. Therefore, idle state energy consumption dominates the results of routing schemes that do not use a duty-cycle mechanism. That is the reason why all the selected protocols presented similar energy consumption results. We have taken that into consideration and evaluated the energy consumption by setting the idle power consumption to zero. With this, although unrealistic, only the energy used by the data dissemination mechanisms will be considered, leaving idle state consumption out of the calculations. Results are shown in Fig. 7(a). Dual-Sink shows slightly better results due to its simple data dissemination mechanism, in which there is always a static sink in the network, resulting in fewer control packets. The eTrail protocol presents similar results to Dual-Sink, however its cluster and trail maintenance mechanisms induce extra overhead. The eTrail without sleeping version is expected to consume more energy than eTrail since no nodes will be put to sleep, and these nodes will participate in cluster configuration, updates and data dissemination in every round.

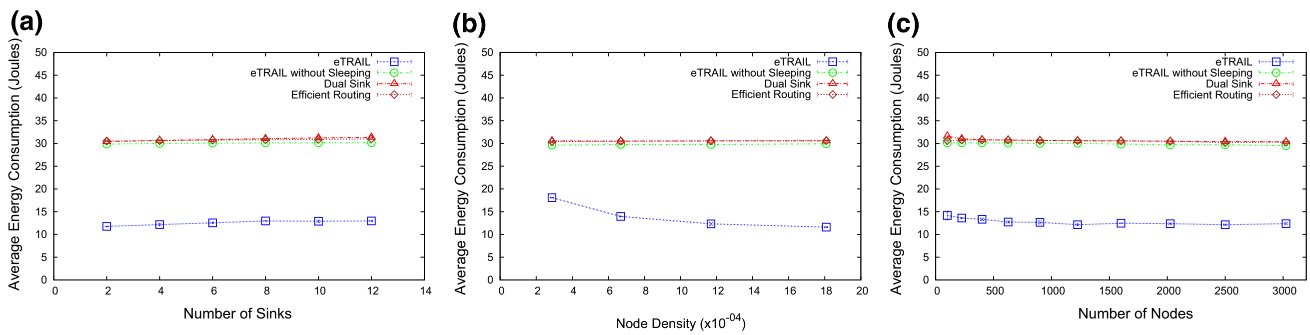

Fig. 6 Average energy consumption per node

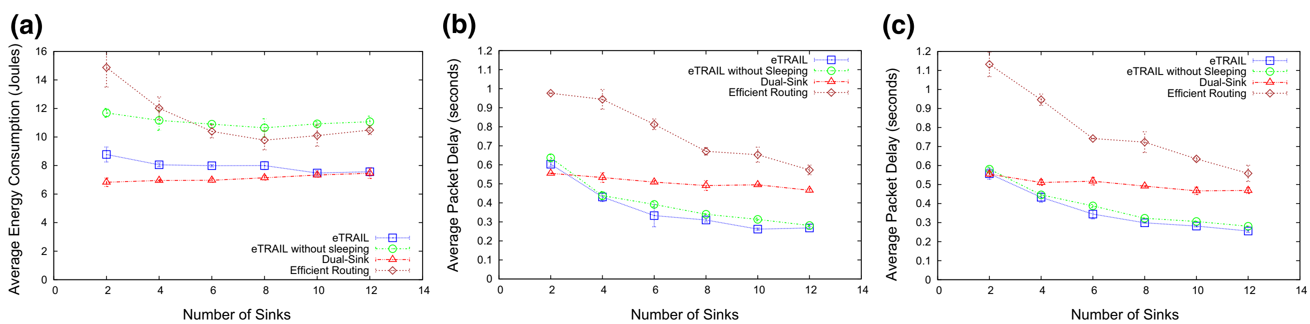

Fig. 7 **a** Average energy consumption per node when idle consumption is set to zero; **b** average delay (4 sources); **c** average delay (10 sources)

4.2 Average data delay

As observed in Fig. 7(b) and (c), the number of source nodes does not have a significant impact on data delay. On the other hand, the number of sinks has a great impact on delay. Since the evaluated protocols are based on flooding and shortest path algorithms, adding mobile sinks reduces data path length between source and sink, which directly reduces packet delay. Dual-Sink is the only protocol that is not affected by the number of mobile sinks. The main reason is the limited hop count used by Dual-Sink when collecting data from nearby nodes. Especially in large-scale WSNs, Dual-Sink does not take advantage of additional mobile sinks. This could be improved by dynamically adjusting the hop count based on network size. Another reason is that Dual-sink uses a static sink, therefore nodes will always have a path to the sink.

As observed in Fig .7, eTrail shows the lowest latency when compared to the selected protocols. Although the trail mechanism in eTrail adds extra nodes to a path, the round duration can be adjusted to accommodate different delay expectations. By simulation experiments, we found that a 50-s trail round period showed a good balance between overhead, energy consumption, and packet delay. Therefore, trails are reconfigured before they become too long in terms of hop count. Conversely, the Efficient Routing protocol showed higher data delay results when compared to the other protocols. In Efficient Routing, nodes farther away from the sink are updated less regularly than nodes closer to the sink in order to minimize mobility management overhead. Hence, sensor nodes located far away from a mobile sink may send data through a high number of hops, which impacts data delay significantly. In other words, Efficient Routing saves energy at the cost of longer routing paths and increased data delay. Since sensor nodes in Efficient Routing only forward sink beacons when the change in cost reaches a certain threshold, some paths far from the ideal shortest paths happen frequently, which induces higher network latency. Regarding the Dual-Sink protocol, many data packets are forwarded to the static sink in the absence of a nearby mobile sink, a fact that induces extra delay.

Since mobile sink mobility is uncontrolled in eTrail, and a mobile sink initiates the rounds, trail length can vary, e.g., the mobile sink may move far away from the cluster, or it may stay stationary. Since a cluster and the current round must have fixed duration, because the dormant nodes will have to wake up for the next roun), the trail age is given in time intervals. Thus, an optimization suggestion would be to use a weighted average of previous travel distance the mobile sink moved between rounds, and find a balance between travel time and overhead induced in a round to build and maintain a cluster and its trail. One could also consider adding metric constraints depending on the application. For instance, if packet delay is important to the application, trail age should be limited to a hop count that delivers packets within a certain delay threshold. However, as mentioned earlier, the trajectory of a mobile sink in eTrail is uncontrolled and not known beforehand, which can affect negatively the attempt to find a sub-optimal trail age or "length", which is tied to the age of cluster and sleeping duration.

4.3 Protocol overhead

This section evaluates and discusses the control overhead of the selected protocols. Since the number of sources and data packet rates of each protocol are the same, the results reflect the number of control packets that are used for building routes, updating mobile sink's location, maintaining data forwarding paths and coordinating data delivery in each protocol. Considering that flat data dissemination protocols are usually sink-oriented, the overhead of the selected protocols are not affected by the number of sources, as depicted in Fig. 8(a) and (b). On the other hand, the number of mobile sinks has a direct impact on protocol overhead: the more mobile sinks present in a network, the more sink location updates are disseminated.

The proposed eTrail protocol induces more overhead than the other protocols, since it handles cluster and trail configuration and updates. However, the overhead is minimized since mobile sinks broadcast beacons to sensor nodes within two hops only. In addition, the trail-based update mechanism does not flood the entire network. Efficient Routing can manage the overhead quite well by setting a cost change threshold at each sensor at the cost of higher packet delay. Dual-Sink has the lowest overhead among all protocols due to the presence of a static sink on the network. This static sink broadcasts its location to the entire network only once at the beginning of the network operation. Meanwhile, the mobile sink of Dual-Sink only plays a role in helping alleviate the hot spot problem, and it only sends out beacons to nearby sensors once in a while to collect data. Although eTrail induces more overhead than the other protocols, it uses this overhead to perform other tasks such as trail generation and sleep-scheduling, which results in better energy savings, reduced packet delays, and satisfactory data delivery success.

4.4 Success data ratio

As observed in Fig. 9(a), eTrail maintains a stable performance around 80 %, which is comparable to Efficient Routing and Dual-Sink. Dual-Sink outperforms the other protocols when only 2 sinks are used in the network. Since Dual-Sink relies on both static and mobile sinks, therefore, sensor nodes always have a path to a static sink when no mobile sink is available. This static sink provides more reliable and stable routes from sources to sink.

Regarding the results of eTrail without the sleeping mechanism, one can conclude that the sleeping scheme affects the delivery success rate mainly because of NO_SLEEP message loss. When a NO_SLEEP message is lost, the node that was supposed to receive this message enters the sleep state after a certain timeout, and a few subsequent data packets are lost until the path recovery mechanism kicks in and finds an alternate route.

Data delivery success was also assessed in terms of node density. As depicted in Fig. 9(b), eTrail is sensitive to node density. In experiments with low density scenarios, eTrail shows delivery ratio of as low as 60 %. This is mainly due to nodes incorrectly switching to sleep state when NO_SLEEP messages are lost. This is also due to the fact that there are only a few or even no nodes replying to HELP messages when a path is broken.

As showed in Fig. 9(c), the number of nodes plays a significant role in data delivery success of flat data dissemination protocols. For small size networks, sinks will frequently visit sensor nodes and data paths are shorter than larger networks. As network size increases, data delivery

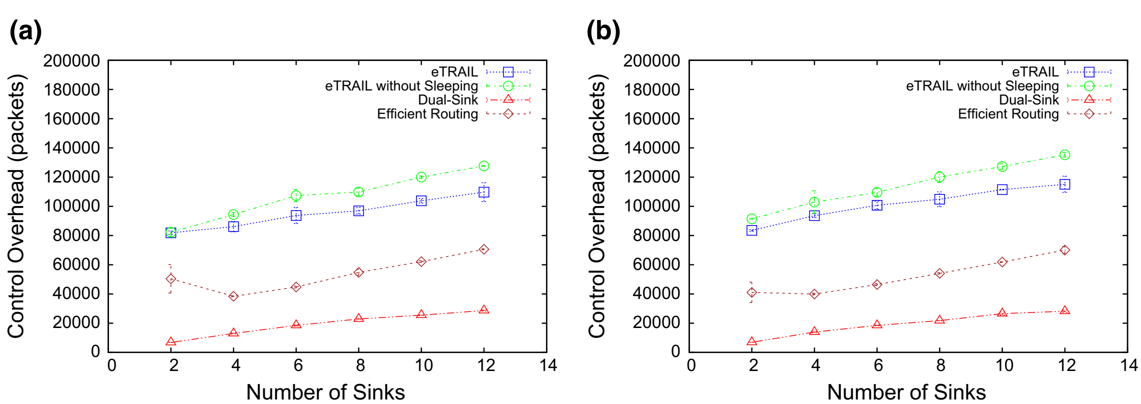

Fig. 8 **a** Control overhead for a network with 4 source nodes; **b** control overhead for a network with 10 source nodes;

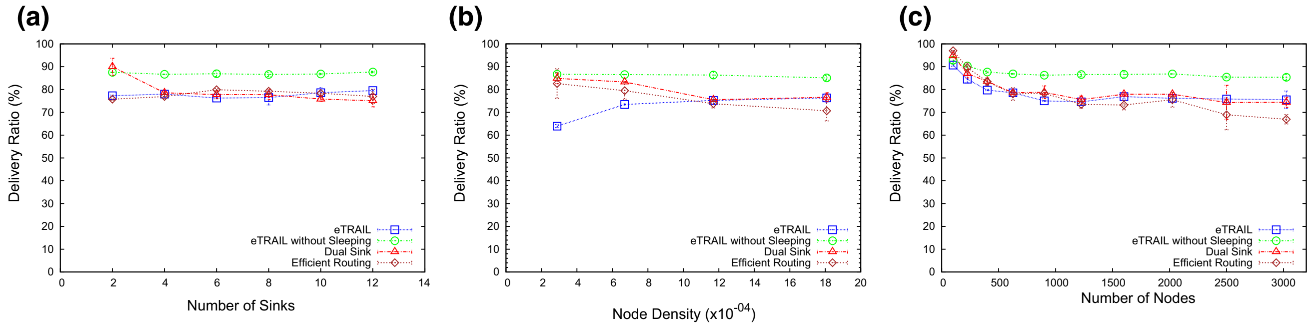

Fig. 9 Data delivery success ratio

success decreases. The probability of a node failing in a long path is higher than in a short path, inducing more path breaks and consequently lower success ratio. All studied protocols present similar results. It is worth noting that eTrail without the sleeping mechanism outperforms the other protocols. This is mainly due to the higher probability of finding an alternative path since all nodes will be awake. The sleeping mechanism in eTrail actually reduces node density within the cluster, which may affect delivery ratio as shown in Fig. 9(b). When nodes are sleeping and a path is broken, fewer nodes will be awake to reply to HELP messages and start the recovery mechanism.

4.5 Network lifetime

In this set of simulation experiments, we evaluate network lifetime performance of each protocol. For these experiments, we set the initial energy of all sensor nodes as low as possible in order to observe energy depletion within the first 1000 s of network operation. In the literature, network lifetime is considered as the elapsed time between the start of network operation and when the first node runs out of energy. Therefore, the initial energy of each sensor is set to 12 Joules. Additionally, four mobile sinks are randomly deployed and ten nodes are randomly selected as data sources. Since eTrail is sensitive to node density, experiments were conducted in two different node density scenarios to better assess its network lifetime. The low density scenario has 576 nodes and the high density scenario has 1600 nodes, both scenarios have nodes deployed over an area of 1200×1200 m. As shown in Fig. 10(a), in a low density scenario, eTrail improves network lifetime significantly due to its sleep-wake mechanism. Simulation results show an improvement of at least 20 % when compared to the selected protocols. Moreover, if one considers the sensor network with 99 to 70 % of alive nodes, i.e., the network is still able to function properly in this range of alive nodes, the network lifetime gain over the selected protocols is even more evident. As can be observed, the selected protocols have most of their nodes dead almost simultaneously. Again, we can conclude that the energy consumption in idle state dictates the lifetime expectancy of protocols that do not use a duty-cycle mechanism or a more efficient MAC protocol.

In the high density scenario, shown in Fig. 10(b), eTrail outperforms the other protocols by around 30 % in terms of network lifetime. Node density plays an important role on energy consumption since more nodes will be able to sleep in more dense areas. These results indicate the positive tradeoff between overhead and energy consumption of the proposed eTrail protocol.

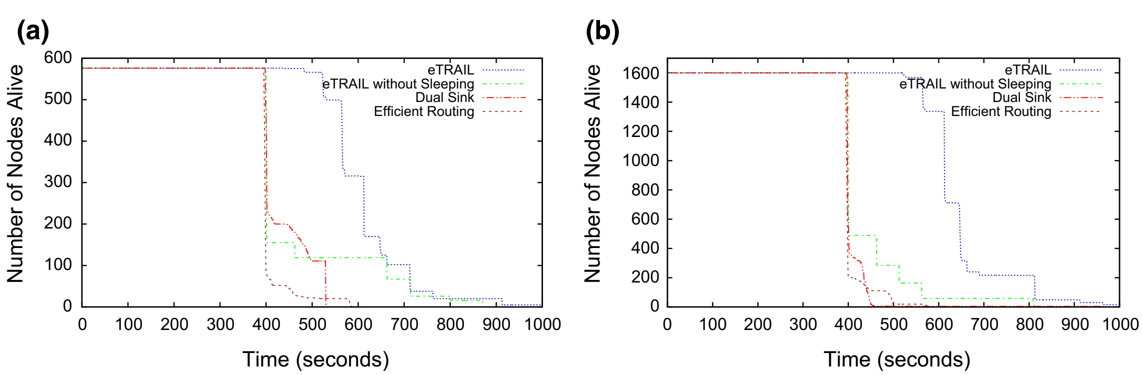

Fig. 10 **a** Network lifetime in a low density scenario. **b** Network lifetime in a high density scenario

5 Conclusions

This research work presented the design, analysis and performance evaluation of a novel data dissemination mechanism intended to improve network lifetime of WSNs by alleviating the problem of high energy depletion of nodes near data sinks. The proposed protocol uses mobile sinks and data "trails" in order to outperform existing data dissemination mechanisms. Experimental results show that eTrail improves on energy savings and network lifetime, while maintaining acceptable delivery ratio when compared to the selected data dissemination mechanisms. According to the experimental evaluations, one can notice that the sleep-wake mechanism dictates the gain in network lifetime. For future work, we plan to investigate how to improve data delivery success in low density scenarios, while improving network lifetime by consider multi-path data delivery. Another line of research we are seeking is the adaptation of the proposed mechanism for hybrid vehicular and sensor networks, where we can take advantage of deterministic paths by predicting the movement of mobile sink on road maps. In addition, duty-cycle enabled sensor nodes combined with data fusion or compressive sensing [16] has the potential to improve network lifetime significantly, with the added bonus of reducing traffic. The integration of compressive sensing into eTrail is left as future work.

References

1. Azad, A. P., & Chockalingam, A. (2006). Mobile base stations placement and energy aware routing in wireless sensor networks. *IEEE on Wireless Communications and Networking Conference (WCNC)*, *1*, 264–269.
2. Basagni, S., Carosi, A., Melachrinoudis, E., Petrioli, C., & Wang, Z. M. (2008). Controlled sink mobility for prolonging wireless sensor networks lifetime. *Wireless Networks*, *14*(6), 831–858.
3. Bhuiyan, M., Wang, G., & Vasilakos, A. (2015). Local area prediction-based mobile target tracking in wireless sensor networks. *IEEE Transactions on Computers*, *64*(7), 1968–1982. doi:10.1109/TC.2014.2346209.
4. Boukerche, A. (2008). *Algorithms and protocols for wireless, mobile ad hoc networks* (Vol. 77). London: Wiley.
5. Chatzigiannakis, I., Kinalis, A., & Nikoletseas, S. (2006). Sink mobility protocols for data collection in wireless sensor networks. In *Proceedings of the 4th ACM international workshop on mobility management and wireless access, MobiWac '06*, pp. 52–59
6. Chen, M., Wan, J., Gonzalez, S., Liao, X., & Leung, V. (2014). A survey of recent developments in home m2m networks. *IEEE on Communications Surveys Tutorials*, *16*(1), 98–114. doi:10.1109/SURV.2013.110113.00249.
7. Di Francesco, M., Das, S. K., & Anastasi, G. (2011). Data collection in wireless sensor networks with mobile elements: A survey. *ACM Transactions on Sensor Networks*, *8*(1), 7:1–7:31.
8. Fodor, K., & Vidács, A. (2007). Efficient routing to mobile sinks in wireless sensor networks. In *Proceedings of the 3rd international conference on wireless internet, WICON'07*, pp. 1–7. ICST, Brussels, Belgium
9. Gandham, S. R., Dawande, M., Prakash, R., & Venkatesan, S. (2003). Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In *Global Telecommunications Conference, 2003 (GLOBECOM '03)* (Vol. 1, pp. 377–381), 1–5 December 2003. IEEE. doi:10.1109/GLOCOM.2003.1258265.
10. Gu, Y., Ji, Y., Li, J., & Zhao, B. (2013). Eswc: Efficient scheduling for the mobile sink in wireless sensor networks with delay constraint. *IEEE Transactions on Parallel and Distributed Systems*, *24*(7), 1310–1320.
11. Gu, Z., Hua, Q.S., Wang, Y., & Lau, F. (2013). Reducing information gathering latency through mobile aerial sensor network. In *Proceedings IEEE on INFOCOM, 2013*, pp. 656–664
12. Hamida, E., & Chelius, G. (2008). A line-based data dissemination protocol for wireless sensor networks with mobile sink. In *IEEE international conference on communications, ICC '08.*, pp. 2201–2205
13. Han, K., Luo, J., Liu, Y., & Vasilakos, A. (2013). Algorithm design for data communications in duty-cycled wireless sensor networks: A survey. *IEEE on Communications Magazine*, *51*(7), 107–113. doi:10.1109/MCOM.2013.6553686.
14. Khan, M. I., Gansterer, W. N., & Haring, G. (2013). Static vs. mobile sink: The influence of basic parameters on energy efficiency in wireless sensor networks. *Computer Communications*, *36*(9), 965–978. Reactive wireless sensor networks.
15. Li, M., Li, Z., & Vasilakos, A. (2013). A survey on topology control in wireless sensor networks: Taxonomy, comparative study, and open issues. *Proceedings of the IEEE*, *101*(12), 2538–2557. doi:10.1109/JPROC.2013.2257631.
16. Liu, X. Y., Zhu, Y., Kong, L., Liu, C., Gu, Y., Vasilakos, A., et al. (2015). Cdc: Compressive data collection for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, *26*(8), 2188–2197. doi:10.1109/TPDS.2014.2345257.
17. Luo, J., & Hubaux, J. P. (2005). Joint mobility and routing for lifetime elongation in wireless sensor networks. In *24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*, Proceedings IEEE (Vol. 3, pp. 1735–1746), 13–17 March 2005. IEEE. doi:10.1109/INFCOM.2005.1498454.
18. Luo, J., Panchard, J., Pirkowski, M., Grossglauser, M., & Hubaux, J. P. (2006). Mobiroute: Routing towards a mobile sink for improving lifetime in sensor networks. *Distributed Computing in Sensor Systems, LNCS*, *4026*, 480–497.
19. Marta, M., & Cardei, M. (2009). Improved sensor network lifetime with multiple mobile sinks. *Pervasive and Mobile Computing*, *5*(5), 542–555.
20. Memsic iMote2 IPR2400 Processing and Sensing Board—Data Sheet (2014).
21. Pazzi, R. W., & Boukerche, A. (2008). Mobile data collector strategy for delay-sensitive applications over wireless sensor networks. *Computer Communications*, *31*(5), 1028–1039.
22. Rault, T., Bouabdallah, A., & Challal, Y. (2013). Wsn lifetime optimization through controlled sink mobility and packet buffering. In *Global information infrastructure symposium, 2013*, pp 1–6.
23. Sengupta, S., Das, S., Nasir, M., Vasilakos, A., & Pedrycz, W. (2012). An evolutionary multiobjective sleep-scheduling scheme for differentiated coverage in wireless sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, *42*(6), 1093–1102. doi:10.1109/TSMCC.2012.2196996.

24. Shin, J. H., Kim, J., Park, K., & Park, D. (2005). Railroad: Virtual infrastructure for data dissemination in wireless sensor networks. In *Proceedings of the 2Nd ACM international workshop on performance evaluation of wireless ad hoc, sensor, and ubiquitous networks, PE-WASUN '05*, pp. 168–174.
25. Turgut, D., & Bölöni, L. (2011). Heuristic approaches for transmission scheduling in sensor networks with multiple mobile sinks. *The Computer Journal*, *54*(3), 332–344.
26. Vass, D., & Vidács, A. (2005). Positioning mobile base station to prolong wireless sensor network lifetime. In *Proceedings of the 2005 ACM conference on emerging network experiment and technology, CoNEXT '05*, pp. 300–301.
27. Vecchio, M., Viana, A. C., Ziviani, A., & Friedman, R. (2010). Deep: Density-based proactive data dissemination protocol for wireless sensor networks with uncontrolled sink mobility. *Computer Communications*, *33*(8), 929–939. doi:10.1016/j.comcom.2010.01.003.
28. Vincze, Z., Vass, D., Vida, R., Vidács, A., & Telcs, A. (2006). Adaptive sink mobility in event-driven multi-hop wireless sensor networks. In *Proceedings of the first international conference on integrated internet ad hoc and sensor networks, InterSense '06*.
29. Wang, G., & Wang., T. (2007). Local update-based routing protocol in wireless sensor networks with mobile sinks. In *Proceedings of the 2007 IEEE international conference on communications, ICC'07*, pp. 3094–3099.
30. Wang, G., Wang, T., Jia, W., Guo, M., & Li, J. (2009). Adaptive location updates for mobile sinks in wireless sensor networks. *Journal of Supercomputing*, *47*, 127–145.
31. Wang, Z., Basagni, S., Melachrinoudis, E., & Petrioli, C. (2005). Exploiting sink mobility for maximizing sensor networks lifetime. pp. 287a–287a.
32. Wu, X., & Chen, G. (2007). Dual-sink: Using mobile and static sinks for lifetime improvement in wireless sensor networks. pp. 1297–1302.
33. Yao, Y., Cao, Q., & Vasilakos, A. (2015). Edal: An energy-efficient, delay-aware, and lifetime-balancing data collection protocol for heterogeneous wireless sensor networks. *IEEE/ACM Transactions on Networking*, *23*(3), 810–823. doi:10.1109/TNET.2014.2306592.
34. Zeng, Y., Xiang, K., Li, D., & Vasilakos, A. (2013). Directional routing and scheduling for green vehicular delay tolerant networks. *Wireless Networks*, *19*(2), 161–173. doi:10.1007/s11276-012-0457-9.

Richard Werner Pazzi is an Assistant Professor at the Faculty of Business and Information Technology, University of Ontario Institute of Technology, Canada. He received his Ph.D. degree in Computer Science from the University of Ottawa, Canada, in 2008. His research interests include fault-tolerant data dissemination protocols for Wireless Sensor Networks and Mobile Computing. He is also active in the areas of Vehicular Ad Hoc Networks, multimedia communications and networked 3D virtual environments. He is the recipient of Best Research Paper Awards from the IEEE International Conference on Communications (ICC 2009), the International Wireless Communications and Mobile Computing Conference (IWCMC 2009), the IEEE Symposium on Computers and Communications (ISCC 2015), and the recipient of Elsevier's Top Cited Article (2005–2010) for his work published in the Journal of Parallel and Distributed Computing (JPDC 2006). He served as co-chair of numerous IEEE and ACM sponsored conferences including ACM DIVANET 2011, ACM Performance Modeling and Monitoring of Heterogeneous Wired and Wireless Networks 2007–2013, and IEEE GLOBECOM 2014 Next Generation Networks Symposium.

Azzedine Boukerche (FIEEE, FEiC, FCAE, FAAAS) is a full professor and holds a Canada Research Chair position at the University ofOttawa (Ottawa). He is the founding director of the PARADISE Research Laboratory, School of Information Technology and Engineering (SITE), Ottawa. Prior to this, he held a faculty position at the University of North Texas, and he was a senior scientist at the Simulation Sciences Division, Metron Corp., San Diego. He was also employed as a faculty member in the School of Computer Science, McGill University, and taught at the Polytechnic of Montreal. He spent an year at the JPL/NASA-California Institute of Technology, where he contributed to a project centered about the specification and verification of the software used to control interplanetary spacecraft operated by JPL/NASA Laboratory. His current research interests include Wireless Ad Hoc and sensor networks, wireless networks, mobile and pervasive computing, wireless multimedia, QoS service provisioning, performance evaluation and modeling of large-scale distributed systems, distributed computing, large-scale distributed interactive simulation, and parallel discrete-event simulation. He has published several research papers in these areas. He served as a guest editor for the Journal of Parallel and Distributed Computing (special issue for routing for mobile ad hoc, special issue for wireless communication and mobile computing, and special issue for mobile ad hoc networking and computing), ACM/Kluwer Wireless Networks, ACM/Kluwer Mobile Networks Applications, and Journal of Wireless Communication and Mobile Computing. He has been serving as an Associate Editor of ACM Computing Surveys, IEEE Transactions on Parallel and Distributed systems, IEEE Transactions on Vehicular Technology, Elsevier Ad Hoc Networks, Wiley International Journal of Wireless Communication and Mobile Computing, Wileys Security and Communication Network Journal, Elsevier Pervasive and Mobile Computing Journal, IEEE Wireless Communication Magazine, Elseviers Journal of Parallel and Distributed Computing, and SCS Transactions on Simulation. He was the recipient of the Best Research Paper Award at IEEE/ACM PADS 1997, ACM MobiWac 2006, ICC 2008, ICC 2009 and IWCMC 2009, and the recipient of the Third National Award for Telecommunication Software in 1999 for his work on a distributed security systems on mobile phone operations. He has been nominated for the Best Paper Award at the IEEE/ACM PADS 1999 and ACM MSWiM 2001. He is a recipient of an Ontario Early Research Excellence Award (previously known as Premier of Ontario Research Excellence Award), Ontario Distinguished Researcher Award, Glinski Research Excellence Award, IEEE CS Golden Core Award, IEEE Canada Gotlieb Medal Award, IEEE ComSoc Expectional Leadership Award, IEEE TCPP Excpetional Leadership Award. He is a cofounder of the QShine International Conference on Quality of Service for Wireless/Wired Heterogeneous Networks (QShine 2004). He served as the general chair for the Eighth ACM/IEEE Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, and the Ninth ACM/IEEE Symposium on Distributed Simulation and Real-Time Application (DSRT), the program chair for the ACM Workshop on QoS and Security for Wireless and Mobile Networks, ACM/IFIPS Europar 2002 Conference, IEEE/SCS Annual Simulation Symposium (ANNS 2002), ACM WWW 2002, IEEE MWCN 2002, IEEE/ACM MASCOTS 2002, IEEE Wireless Local Networks WLN 0304; IEEE WMAN 0405, and ACM MSWiM 9899, and a TPC member of numerous IEEE and ACM sponsored conferences. He served as the vice general chair for the Third IEEE Distributed Computing for Sensor Networks (DCOSS) Conference in 2007, as the program cochair for GLOBECOM 20072008 Symposium on Wireless Ad Hoc and Sensor Networks, and for the 14th IEEE ISCC 2009 Symposium on Computer and Communication Symposium, and as the finance chair for ACM Multimedia 2008. He also serves as a Steering Committee chair for the ACM Modeling, Analysis and Simulation for Wireless and Mobile Systems Conference, the ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, and IEEE/ACM DSRT.

Robson Eduardo De Grande is currently the Network Manager of the NSERC DIVA Strategic Research Network and a Senior Research Associate at the School of Electrical Engineering and Computer Science, University of Ottawa, Canada. He received his Ph.D. degree at the University of Ottawa in 2012 and his M.Sc. and B.Sc. degrees in Computer Science from the Federal University of Sao Carlos, Brazil in 2006 and 2004, respectively. He served as TPC Chair of IEEE/ACM DS-RT 2014. His research interests include large-scale distributed and mobile systems, Cloud computing, high performance computing, performance modeling and simulation, computer networks, vehicular networks, and distributed simulation systems.

Lynda Mokdad received her Ph.D. in computer science from the University of Versailles, France in 1997. She was associate professor at University Paris-Dauphine from 1998 to 2009. She is currently full professor at University Paris-Est, Créteil since 2009. Her main research interests are about performance evaluation techniques and applications in wired, mobile and wireless networks and in software technologies as Web services. She has published around 60 papers in journals and conferences. She is recipient of the best paper awards of IEEE International Conference on Communications and Information Technology (ICCIT 2011) and IEEE International Conference on Communications (ICC 2011). She has served as technical committee for more than 20 international IEEE/ACM conferences and workshops including Mascots, GlobeCom, MSWIM, LCN, etc. She is a member of IEEE and ACM and have served as a program co-chair of International Conference on Communications (ICC 2015) and (ICC 2013), Global Communications Conference (GlobeCom 2012); 5th and 6th IEEE International Workshop on Performance Management of Wireless and Mobile Networks (P2MNET 2009 and 2010); and 11th international Workshop on Wireless local Networks (WLN 11). She is the founder of IEEE performance evaluation of communications in distributed systems and Web based service architectures (PEDISWESA) which arrives this year at the sixth. She serves as editor of Wiley International Journal of Communication Systems (IJCS) and Wireless Communications and Mobile Computing (WCMC). She was secretary and vice chair of IEEE Communication software (CommSoft) and she is currently serving as a chair of this committee.