EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Anchor-based routing protocol with dynamic clustering for Internet of Things WSNs

Catalina Aranzazu-Suescun and Mihaela Cardei*

Abstract

Wireless sensor networks (WSNs) have many applications in climate monitoring, fire detection, smart homes, and smart cities, as well as detecting and monitoring dynamic, spatio-temporal events such as storms, traffic, and animal movement. WSNs are an important component of the Internet of Things (IoT), and they play an important role in monitoring and collecting surrounding context and reporting the sensed data. In this paper, we consider a WSN deployed to monitor spatio-temporal events and a mobile sink which acts as the gateway between the IoT and the WSN. Data gathering is a challenging problem in WSNs and IoT since the algorithms have to be energy-efficient and the information has to be available to the end user in a timely manner and without redundancies.

In this paper, we propose an improvement of the reactive, anchor-based routing protocol with constrained flooding and dynamic clustering from Aranzazu-Suescun and Cardei (Spatio-temporal event detection and reporting in mobile-sink wireless sensors networks, 2017). We propose a new event-based clustering mechanism and a new dynamic clustering technique. This improvement results in reduced energy consumption and a higher number of packets processed successfully by the sink compared to Aranzazu-Suescun and Cardei (Spatio-temporal event detection and reporting in mobile-sink wireless sensors networks, 2017). Data collected by the mobile sink are shared to the end users through the IoT infrastructure. We conducted simulations using WSNet, an event-driven simulator for WSNs. We measured various performance metrics, such as the average residual energy, the number of active clusters, and the percentage of events processed successfully by the sink.

Keywords: Wireless sensor networks, Internet of things, Mobile sink, Data gathering, Dynamic clustering, Energy efficiency

1 Introduction

Interconnecting devices to generate a “smart” environment is the basic concept in the Internet of Things (IoT), and the sensor-actuator-internet is the framework to achieve this smart environment [5]. Wireless sensor networks (WSNs) have been widely used in many IoT applications due to ubiquitous sensor devices [3, 5, 19]. Extensive research activities in the areas of security [18, 20], topology [24], synergies with other technologies [17], and energy consumption [5] in WSNs for IoT have been conducted recently.

WSN data collection and event reporting are important research topics in the IoT. The IoT is a worldwide network where all devices are interconnected and information has to be available fast and efficiently, thus redundancies and useless information have to be eliminated. A key approach for efficient interconnection is to give devices a “smart” behavior where they can communicate, process information, and take decisions without human intervention [13].

Designing energy-efficient communication protocols in WSNs is vital. Most of the sensors are battery powered, and sometimes it is impractical or infeasible to replace or recharge the nodes. In addition, WSNs may be deployed in areas where access is difficult and monitoring has to be done over a long period of time. Due to the power constraint characteristic of WSNs, the applications using WSNs employ different mechanisms to minimize the energy consumption in order to prolong the network lifetime [1].

* Correspondence: mcardei@fau.edu
Florida Atlantic University, Boca Raton, FL 33431, USA

Depending on the application, events can change their shape, can appear/disappear (e.g., fire events), and can move (e.g., storms such as hurricanes, tornadoes). These events are called spatio-temporal events or dynamic events [23]. Spatio-temporal events have the characteristic that their shape, size, and speed of movement can change over time. Different mechanisms have been developed by researchers in order to save energy in the network. One mechanism is using clustering for data collection. Cluster heads are in charge of collecting data from its cluster members, aggregate it, and send it to the sink [2, 4, 6]. Also, if events are sparse and expected to occur sporadically, then a reactive protocol could be used. In this case, when there is no event, no energy is wasted to maintain an updated path to the sink [7, 8, 16].

In this paper, we improve the solution proposed in [1], where the authors address the problem of detecting and reporting a composite mobile event to a mobile sink. We design a new event-based clustering mechanism and we revised the dynamic clustering technique. This improvement results in reduced energy consumption and a higher number of packets processed successfully by the sink compared to [1]. The rest of the paper is organized as follows. Section 2 describes the design methods of our routing protocol and summarizes the experiments. Section 3 introduces the state of the art in monitoring spatio-temporal events. Section 4 presents the event model. Section 5 describes the problem definition. In Section 6, we present our anchor-based routing protocol for spatio-temporal event detection and reporting. The performance of our algorithm is illustrated in Sections 7 and 8, where we conduct simulations using WSNet [22]. The conclusions are stated in Section 9.

2 Methods/experimental

In this paper, we propose an energy-efficient and distributed routing mechanism for event detection and reporting to a mobile sink. We consider that events can be mobile, thus the set of sensor nodes which detect a certain event change over time. To deal with the sink and event mobility, we use several important techniques in our algorithm design. First, we use a dynamic anchor mechanism. A set of anchors may be active at a certain time, and this set may change over time depending on the sink movement. Another important technique is the use of a dynamic, event-based clustering, which allows the collection of information about the moving events. The cluster (or clusters) monitoring a certain event may change over time, according to the movement of the event. This dynamic clustering mechanism reduces the number of event-reporting messages sent to the sink, thus the overall energy consumed by the network is decreased.

We conduct simulations using the WSNet [22] network simulator and measure various performance metrics such as the average residual energy, the number of clusters which are active, and the percentage of composite events processed successfully by the sink. Simulation results show that the newly proposed mechanism has the best performance in terms of energy efficiency and percentage of events detected successfully by the sink.

3 Related works

According to [23], spatio-temporal events can be diffuse. This means that the events can spread in space similar to a fire or they can move similar to a tornado. The authors use extended linear-chain conditional random fields, which are undirected graphs that use a specific group of features to encode a conditional probability distribution. The authors incorporate temporal constraints to a spatial field in order to determine the spatio-temporal dependencies among observations and events in the network. In this way, the sensors can determine if the event is diffuse or whether it is moving. In our work, the events can be fixed or moving, and we address the problem of reporting the event detection to the sink.

An example of using WSNs for IoT is illustrated in [12]. An indoor distributed sensor system with an intelligent processing unit is used to track a fire. When sensor reading exceeds a given threshold, the sensor node sends its reading and location to the base station (or processing unit). The base station calculates a function $f(x, y, z)$ and its gradient using the coordinates x, y, and z of the node. The results are plotted as vectors. The readings closer to the center of the event will have bigger vectors. The base station sends a “start sensing process message” to the 1-hop neighbors of the border of the event nodes, to continue tracking the movement of the fire. The base station can also calculate the direction of the event movement using the relative weighting of the i, j, and k components of the gradient. Our work is different in several ways. We are not restricted to indoor detection, both the sink and the event are mobile in our case, and we use energy-efficient techniques such as clustering.

Article [11] presents an IoT mechanism for detecting a forest fire using artificial neural network. A multi-layer back-propagation artificial neural network algorithm is used to detect multiple criteria that depend on specific attributes of a forest fire (e.g., temperature, radiation, and light). The neural network is feeded with environmental information collected by heterogeneous sensor nodes, and the correct alarm decisions. With this information, the system is capable of learning and making decisions when new information is feeded in the system. In our work, we assume that the detection is performed by the sensor nodes, and we use energy-efficient techniques to reduce the energy consumed by the WSNs.

WSNs can be used in IoT applications for monitoring environmental changes, people, and animal movements.

Article [15] presents an IoT application developed to monitor the Australian Great Barrier Reef using big data analytics and a hierarchical heterogeneous sensor network. Sensor nodes collect environmental data from the ocean such as salinity, temperature and level of chlorophyll. The sensor nodes collect information, then aggregate, and send it to a buoy. The buoys are connected to poles which are in charge of sending the aggregated data to the base station. The elliptical summaries anomaly detection (ESAD) algorithm uses artificial intelligence to detect anomalous patterns based on the data collected. The algorithm measures data dissimilarities, forms clusters of similar information, and extracts the anomalous clusters. This article has a very important practical applicability, and it is different from our work since it assumes a certain hierarchical infrastructure. Works [9, 10] also deal with monitoring diffuse events. In this case, the sensor network is used to detect the boundaries of the event. A sensor detecting an event (also called "affected" node) checks whether its neighbors detect the event too. If all neighbors are affected as well, then the node proclaims itself a non-boundary node. But if at least one neighbor did not detect the event, then the node broadcasts the result of a function computed based on the event detected and the number of neighbors affected. The node then waits to receive the evaluation of the same function from its neighbors. The node with the smallest function value becomes a boundary node. Then, the boundary node sends the information to the sink using a multi-hop routing protocol. The process is repeated periodically to track the changes in the size of the event. In our work, the algorithm continuously monitors and reports the event, not only the changes in the boundary of the event.

Article [14] proposes an algorithm that uses probabilistic graphical models to detect spatio-temporal events. The authors use a graph representation of the nodes, where every node is a random variable and the edges are the probabilistic interaction between them. The process of detection and reporting has two phases. In the first phase, a temporal dependency modeling of the events is done using a first-order Markov chain model. The nodes predict the next state of their sensed values (cold, mild, hot) based on their own information. The second phase is the spatial dependency model of the event that uses Markov random fields. In this case, the nodes predict their sensing values using the information of its neighbors. Every time a node senses an event, it informs its neighbors so that they can start the process of detecting the event. The detection mechanism used in our case is different, see Section 4, and we are also concerned with employing energy-efficient techniques.

To track a diffuse event in IoT, article [21] uses a Bayesian trust model and a network consisting of a grid of nodes. Each node has a virtual cluster, the node being the center of the cluster. Its members are the neighbors within communication range. The node calculates a trust index with a probability density function. If the value of this trust index is less than some predefined threshold, then the node readings do not match one or more readings of its neighbors. To adjust the trust index, the node uses a Bayesian filtering procedure to incorporate neighbors' readings to its own readings. The result of this procedure is $e = 1$ or $e = 0$, where $e = 1$ means that the node is detecting the event and $e = 0$ means that the event is not detected. Every time period T, the node sends to the sink its Id and the value of e. Based on the information received from the nodes, the sink reconstructs the map of the network and the progression and size of the event. In our case, we assume that the nodes are randomly deployed rather than in a grid. We developed an event-based clustering, where only the nodes detecting the event are involved in reporting the event.

In this paper, we propose a reactive anchor-based routing protocol with constrained flooding and dynamic clustering for spatio-temporal event detection and reporting. Our work improves on the solution presented in [1] and which will be detailed in Section 6.

4 Description of the event model

We define an event as an observable occurrence of a phenomenon during some specific time in a specific area. We use the concepts of *atomic events* and *composite events*, given in [4]. An atomic event is triggered when a single sensing value (or attribute) exceeds some given threshold and it is denoted by $e(t, s, R)$ where t is the time when the event occurs, s is the location of the event, and R is a logical expression defining the conditions when the event occurs. To detect complex events, we use composite events, where variations in different attributes are detected. A composite event is denoted by the following:

$$E((e_1, \delta_1), (e_2, \delta_2), \ldots, (e_k, \delta_k), C_t, C_s, \delta) \\ = (R_1 \wedge R_2 \wedge \ldots \wedge R_k \wedge C_t \wedge C_s, \delta)$$

where e_i, $i = 1$ to k, is an atomic event. δ_i with $0 \le \delta_i \le 1$ is the confidence of e_i, indicating the probability of E occurring when e_i occurs. C_t is the constraint on atomic events' times $t_1, t_2, \ldots, t_k$. Cs is the constraint on atomic events' locations $s_1, s_2, \ldots, s_k$. The confidence δ of the composite event is defined as $\delta = \delta_1 + \delta_2 + \ldots + \delta_k$, and it is expected to satisfy the property $\delta_1 + \delta_2 + \ldots + \delta_k = 1$ [4].

We consider spatio-temporal events similar to [1, 23]. Our event moves with a certain speed, but it maintains the same circular shape and moves on a random path, see Fig. 1.

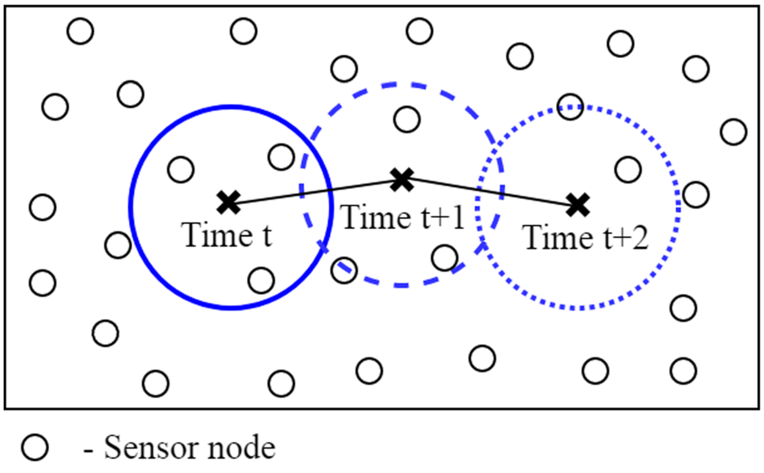

Fig. 1 Spatio-temporal event

5 Problem definition

We consider a heterogeneous WSN consisting of n nodes N_1, N_2, …, N_n and one mobile sink S. The nodes are randomly deployed in an area A. The nodes have the same communication range R_c and the same initial energy E_{init}. We consider a mobile sink with a communication range R_c and infinite energy. Each node has one or multiple sensing components from the set $\{s_1, s_2, \ldots, s_m\}$.

Each sensing component can be used to detect an atomic event for that attribute [1]. Table 1 shows the main notations in the paper. We consider the following assumptions for our work:

Table 1 Notations

E	Composite event
δ	Confidence of the composite event
e_i	Atomic event i
δ_i	Confidence of atomic event i
n	Number of nodes
m	Maximum number of sensing components
T	Convergecast tree rooted at S
$T_{cluster}$	Cluster tree rooted at CH
N_j	Node j, $1 \le j \le n$
$N_j.\{s_{j1}, s_{j2}, \ldots, s_{jk}\}$	Sensing components of node N_j, $1 \le k \le m$
$N_j.E_{\text{residual}}$	Residual energy of node N_j
$N_j.tp$	Parent of node N_j in T
$N_j.cp$	Parent of node N_j in $Tcluster$
R_c	Node communication range
A	Deployment area
$A.L$	Length of the side of the area
E_{init}	Initial energy of each node

- The nodes have synchronized clocks, so we do not deal with node synchronization in this work.
- The network is heterogeneous, that means sensor nodes may be equipped with different sensing components. This occurs since nodes may be manufactured with different sensing capabilities, some nodes may have purposely turned off some sensing components to save energy, and some sensing components may fail over time.
- Events are fixed or mobile.
- Nodes are resource constrained in terms of power, bandwidth, memory, and computing capabilities.

5.1 Problem definition—spatio-temporal composite event detection and reporting (STCEDR) in mobile-sink WSNs

Given a WSN deployed in an area A, consisting of n nodes with different sensing components from the set $\{s_1, s_2, \ldots, s_m\}$ and a mobile sink S, designing an energy-efficient and distributed algorithm for detecting and reporting a spatio-temporal composite event E inquired by the sink S, the composite event E is defined using atomic events corresponding to the sensing components $\{s_1, s_2, \ldots, s_m\}$.

6 ARER—anchor-based routing protocol for event reporting in mobile-sink WSNs

In this section, we propose ARER, anchor-based routing for event reporting, an improvement of the anchor-based routing protocol with constrained flooding and dynamic clustering presented in [1]. The improvement has two components: a new event-based clustering mechanism and a new rule for dynamic clustering, which will be described later in this section. These mechanisms result in a reduced energy consumption and an increased percentage of composite events processed successfully by the sink, which will be illustrated in Section 8.

Figure 2 shows the main phases of the protocol. In *phase 1* the sink S selects the first anchor A_1 using the following mechanism. S broadcasts *FindClosestNode* and the nodes in range reply with their ID and residual energy after a small delay. The sink chooses the closest node based on the signal strength, and in case of a tie the residual energy and the smallest ID are used. The sink S then sends the request for monitoring the composite event E to A_1, who is in charge of flooding the *CompositeEventRequest*(S, A_1, E, δ_{th}, *hops*) through the network, where E is the composite event and δ_{th} is the threshold parameter for the composite event.

As the message floods the network, a convergecast tree T is formed, where A_1 is the root. Each node N_j that receives the message will set-up its parent N_j *.tp* to the sending node, will increment the *hops* field of the message, and will resend *CompositeEventRequest*(N_j, E, δ_{th}, *hops*). At the end of this phase, each node has a pointer to its parent in the convergecast tree T rooted at A_1.

In *phase 2*, the nodes that satisfy the location requirement C_s and are equipped with sensing components needed to detect one or more atomic events e_1, …, e_k, start monitoring the event for a time duration C_t.

In *phase 3*, one or more nodes start detecting the event. The nodes form one or more clusters. Each cluster selects a cluster head (CH) and initiates the mechanism for event reporting. Our *event-based clustering* is described in Section 6.1.

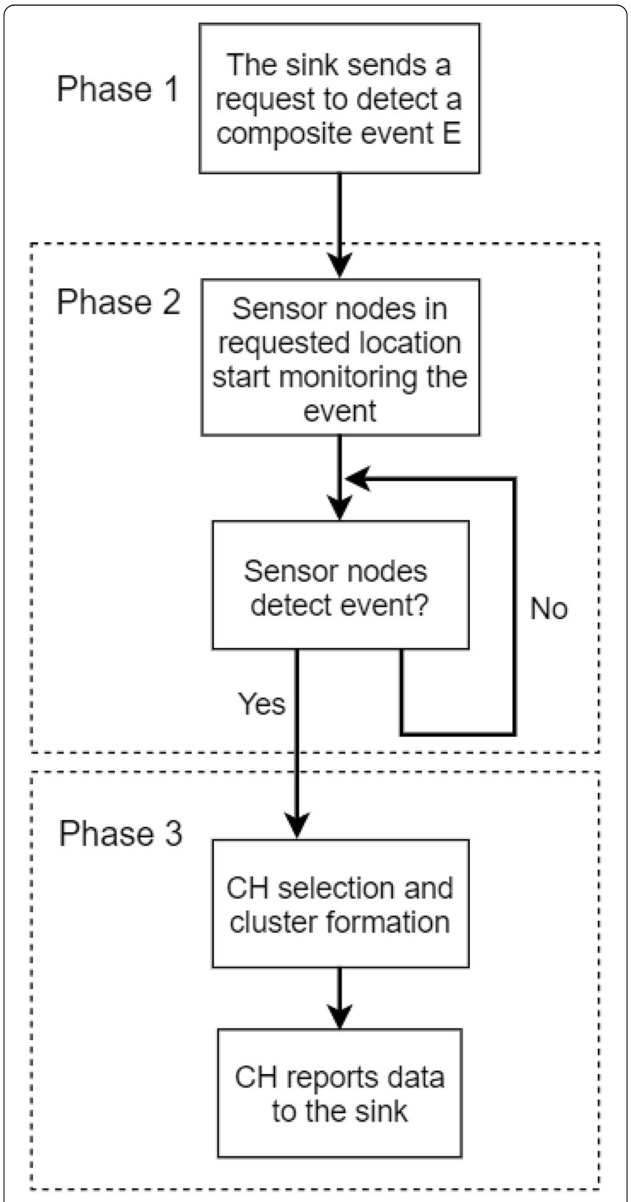
Phase 1
The sink sends a request to detect a composite event E
Phase 2
Sensor nodes in requested location start monitoring the event
Sensor nodes detect event?
No
Yes
Phase 3
CH selection and cluster formation
CH reports data to the sink

Fig. 2 Main phases of the ARER protocol

6.1 Event-based clustering

A node can become CH only if it detects at least one atomic event and if its residual energy is larger than a predefined threshold. Based on its residual energy and ID (used to break ties between nodes with the same residual energy), a node proclaims itself CH and sends a message *NewCluster* over h_{cluster} hops. This message contains the ID of the CH and event ID. All the nodes within the h_{cluster} hops distance join the cluster and resend the message as long as the number of hops is less than h_{cluster}. As the *NewCluster* message is forwarded, a cluster tree T_{cluster} rooted at the CH is formed. The nodes will not send any ack message back to the CH.

Depending on the size of the event and the number of hops in the *NewCluster* message, more than one cluster may form. The CH receives atomic events from its cluster members, and as messages are sent from cluster members to the CH, aggregation is performed.

6.2 Event reporting

The event report messages flow from the nodes to the CH, from the CH to A_1 along T, and from A_1 to S. Since S is mobile, A_1 sends beacons (or data) periodically to the sink to maintain the communication with S. If S does not hear a beacon (or data) from A_1 for α periods, where α is an input parameter, then S selects a new anchor A_2 as follows. S broadcasts a message *NewAnchorRequest*(S, A_1). If a node N_j receives both A_1's beacons (or data) and S's message, then it waits a time based on the signal strength of the message *NewAnchorRequest* and sends a message *NewAnchorReply*(S, A_1, N_j) to the sink. The node N_j from which the sink receives the first reply becomes the second anchor.

If S moves out of the range of A_2, then the process repeats and a new anchor A_3 is selected, see Fig. 3a. In this case, the data travels from the node to A_1, A_2, A_3, and then the sink S, respectively. After the maximum number of anchors β is reached, the anchor selection process resets, that means a new anchor A_1 is selected.

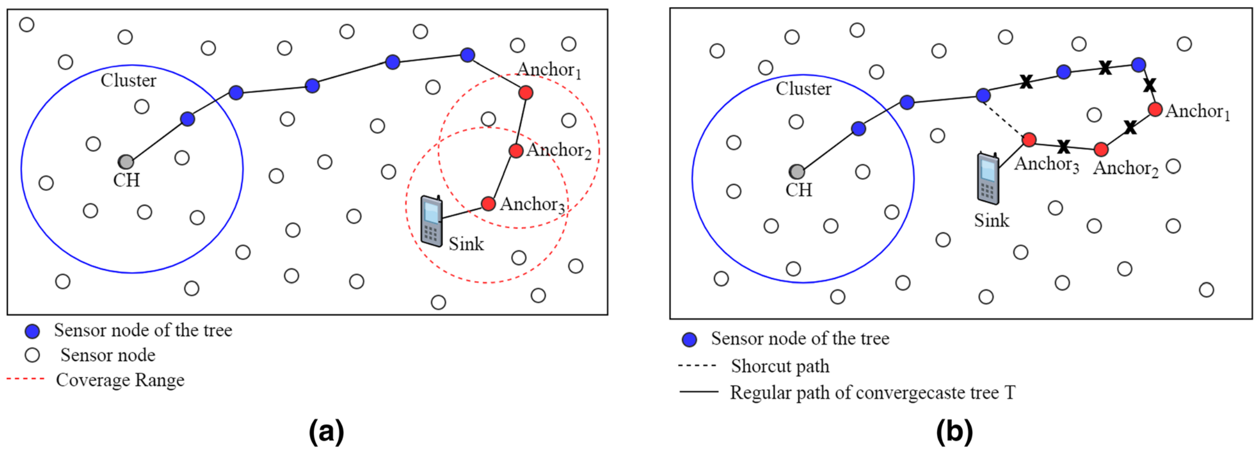

Fig. 3 Example showing **a** the new anchor selection and **b** the shortcut mechanism

To save energy and to reduce the event-reporting delay, we implement a *shortcut* mechanism. If a node N_j in T receives beacons from the last anchor, then N_j stores this anchor as its parent N_j *.tp*. Then N_j sends data directly to the last anchor instead of sending it through the rest of the path. This mechanism is illustrated in Fig. 3b.

To address the case when events cease to exist, a time-out procedure is implemented. If no data (event reports) are sent for a duration γ, then the fields N_j *.tp* are marked obsolete. Similarly, the anchor status of a node becomes obsolete if no data is forwarded toward the sink for a duration γ. Thus, our data collection mechanism is *reactive*, where the data collection infrastructure forms and is maintained only when there are events to report. If the parent field is obsolete, then a CH has to find a path to reach the sink S. The CH broadcasts *RouteRequest*. When S receives the message, it selects an anchor A_1 using the mechanism described previously. A_1 then floods the whole network with a *RouteReply* message which contains the parameters of the composite event. As the *RouteReply* message is sent, a convergecast tree rooted at A_1 is formed. A_1 floods the reply message since often times more clusters are formed, and in this way we avoid other *RouteRequest* messages being initiated by other CHs.

Note that some nodes might have the *tp* attribute active, while others may have this attribute obsolete. To reduce the energy consumption, a "constrained flooding" algorithm is proposed, which is basically an incremental ring-search mechanism. Rather than flooding the whole network, first, the CH broadcasts the *RouteRequest* message to h hops. If a node N_j with an active *tp* attribute receives the message, then it sends back a *RouteReply* message to the CH which contains the number of hops to A_1 or the number of hops to the last anchor if the shortcut mechanism was applied. If the CH receives more than one *RouteRequest*, then it sets its *tp* attribute to the node that has the smallest number of hops to the sink. If no *RouteRequest* is received, then the CH increases the number of hops, and after a number of unsuccessful iterations it floods the network to find a path to the sink.

6.3 Dynamic clustering

When spatio-temporal events move, additional clusters are formed to report the event to the sink S. As the event moves, some of the clusters (or parts of clusters) may not detect the event, while new nodes start detecting the event. These new nodes will form new clusters.

When a new cluster is formed, the CH sends a *NewCluster* message, as described previously in *phase 3*. The nodes which do not belong to any cluster and detect atomic events join the new cluster.

To reduce the number of clusters active in the network, a *dynamic clustering* mechanism is proposed. A node N_j which belongs to a cluster and receives a *NewCluster* message for an event with the same ID, but from another CH, joins the newer cluster. Actually, if more *NewCluster* messages are received by a node N_j for the same event ID, then N_j joins the newest cluster using a time-stamp field in the *NewCluster* message.

A node N_j belongs to a cluster and acts as a relay node for prior clusters in which it belonged. N_j has a table containing the parent node for both current and prior clusters. Let us assume that N_j was a member of the cluster C_1, and it is now a member of C_2. N_j sends its data reports to the CH of C_2. At the same time, N_j acts as a relay node for the nodes transmitting data to the CH of C_1. The data reporting messages are sent to the CH along the paths in the cluster tree, as explained previously.

The dynamic clustering technique is expected to reduce the number of event-reporting messages circulating from CHs to the sink, since older clusters will have no event reporting sooner. Then, less energy is consumed.

Message complexity is discussed next. The number of messages exchanged for phase 1 (see Fig. 2) is $O(n)$, where

n is the number of sensor nodes. In phase 2, no messages are exchanged. In phase 3, each cluster formation has message complexity $O(n)$, and as the event moves, more clusters may form. Thus the overall message complexity for clustering is $O(n^2)$. CHs report data to the sink following the parent path in the convergecast tree T, with message complexity $O(n)$. The overall message complexity of the protocol is $O(n^2)$.

7 Simulation specifications

We compare the performance of three algorithms:

– *ASC*(CF), which is the protocol proposed by *A*ranzazu-*S*uescun and *C*ardei in [1], the option that uses only the constraint flooding (CF)
– *ASC*(CF&*DC*), which is the protocol proposed in [1], the option that uses both the constraint flooding (CF) and the dynamic clustering (DC) mechanisms
– *ARER* (CF&*DC*), which is the protocol presented in Section 6

We conduct simulations using WSNet [22], an open-source event-driven simulator for WSNs. WSNet uses the object-oriented $C++$ language, Linux operating system, and provides support for energy model and event modeling. Tables 2, 3, 4, and 5 present the main parameters used in the simulations.

Our WSN is deployed into a square area with side length $A.L = 1100$ m, with a sink S located in the middle of the right side at the beginning of the simulation.

The WSN has $n = 3125$ nodes randomly deployed in the area A. Each node is equipped with a random number between 1 and 5 of sensing components. We define a composite event with five atomic events shown in Table 3. The composite event that the sensor network monitors is defined as: $E((e_1, 0.35), (e_2, 0.1), (e_3, 0.15), (e_4, 0.3), (e_5, 0.1), t_s, A, \delta)$, where t_s is the simulation time after the request is sent by the sink S and A is the deployment area.

The initial energy of each node is $E_{\text{init}} = 1\,Joule$. To measure the energy consumption in the simulations, we use the energy model from LEACH [6]:

$$E_{Tx}(l,d) = E_{\text{elec}} \times l + E_{\text{amp}} \times l \times d^2$$
$$E_{Rx}(l) = E_{\text{elec}} \times l$$

where $E_{\text{elec}} = 50$ nJ/bit, $E_{\text{amp}} = 100$ pJ/bit/m^2, and d is the distance between nodes.

A node can become a CH if its residual energy exceeds the threshold of 900 mJ. In the simulations, we assume that events have a circular area. Events move using the random walk model with speeds indicated in the Table 5.The center of the event is generated randomly. We consider three types of events

– *small events*, where event radius is 45 m
– *medium events*, where event radius is 200 m
– *large events*, where event radius is 400 m

The simulation time is 1 h. Nodes detecting an event send data messages to the sink S periodically, every 5 s. In the ASC algorithms, cluster membership becomes obsolete after 10 periods. A composite event occurs if the confidence exceeds the threshold value 0.75. We run each simulation scenario 10 times using different seed values and report the average values in the graphs.

8 Results and discussion

Figure 4 shows the average residual energy of the network for $n = 3125$ nodes, $A.L = 1100$ m, medium size events, and average sink speed 5 m/s. We vary the average event speed to 1 m/s, 9 m/s, and 18 m/s, respectively. There is only one continuous event in all the experiments and the duration of the event is 25% and 75% of the total time of the simulation. For example, ASC(CF)(25%) means that the duration of the event is 25% of 1 h, which is 15 min.

In all the experiments, ARER(CF&DC) consumes less energy than the ASC protocols. Both algorithms with dynamic clustering (DC) consume less energy than ASC(CF). This points out the benefits of dynamic

Table 2 Simulation parameters

Simulation time	1 h
Antenna type	omnidirectional
MAC layer	802.11
E_{init}	1 J
Node communication range R_c	100 m
Packet length	132 bytes
Confidence threshold δ_{th}	0.75

Table 3 Types of sensors used

Sensor type	Confidence	Threshold
Temperature	0.35	150
Pressure	0.1	50
Humidity	0.15	10
Smoke	0.3	100
Light	0.1	80

Table 4 Sink speed

Average speed (m/s)	Maximum speed (m/s)
1	2
5	10
12.5	25

Table 5 Event speed

Average speed (m/s)	Maximum speed (m/s)
1	2
9	13
18	30

clustering. Since nodes join new clusters as events move, fewer clusters are involved in data reporting. ARER(CF&DC) is more energy-efficient than ASC(CF&DC) due to the new clustering mechanism which omits acknowledgment messages. A higher speed of the event results in more clusters, thus the residual energy decreases.

Figure 5 shows the residual energy of the network for $n = 3125$ nodes, $A.L = 1100$ m, medium-size events, and average event speed of 9 m/s. The average sink speed is 1 m/s, 5 m/s, and 12.5 m/s, respectively. When the sink moves faster and events are present, new anchors have to be selected to maintain the communication between the clusters and the sink. When the maximum number of anchors is reached, a new convergecast tree is built. Therefore more energy is spent on this process, more frequently.

In Fig. 6, the average sink speed is 5 m/s, the average event speed is 9 m/s, and the number of nodes is $n = 3125$. Results are measured for small, medium, and large events, with one event in the network. More energy is spent by the network on data reporting for larger events since more clusters are formed and report data to the sink.

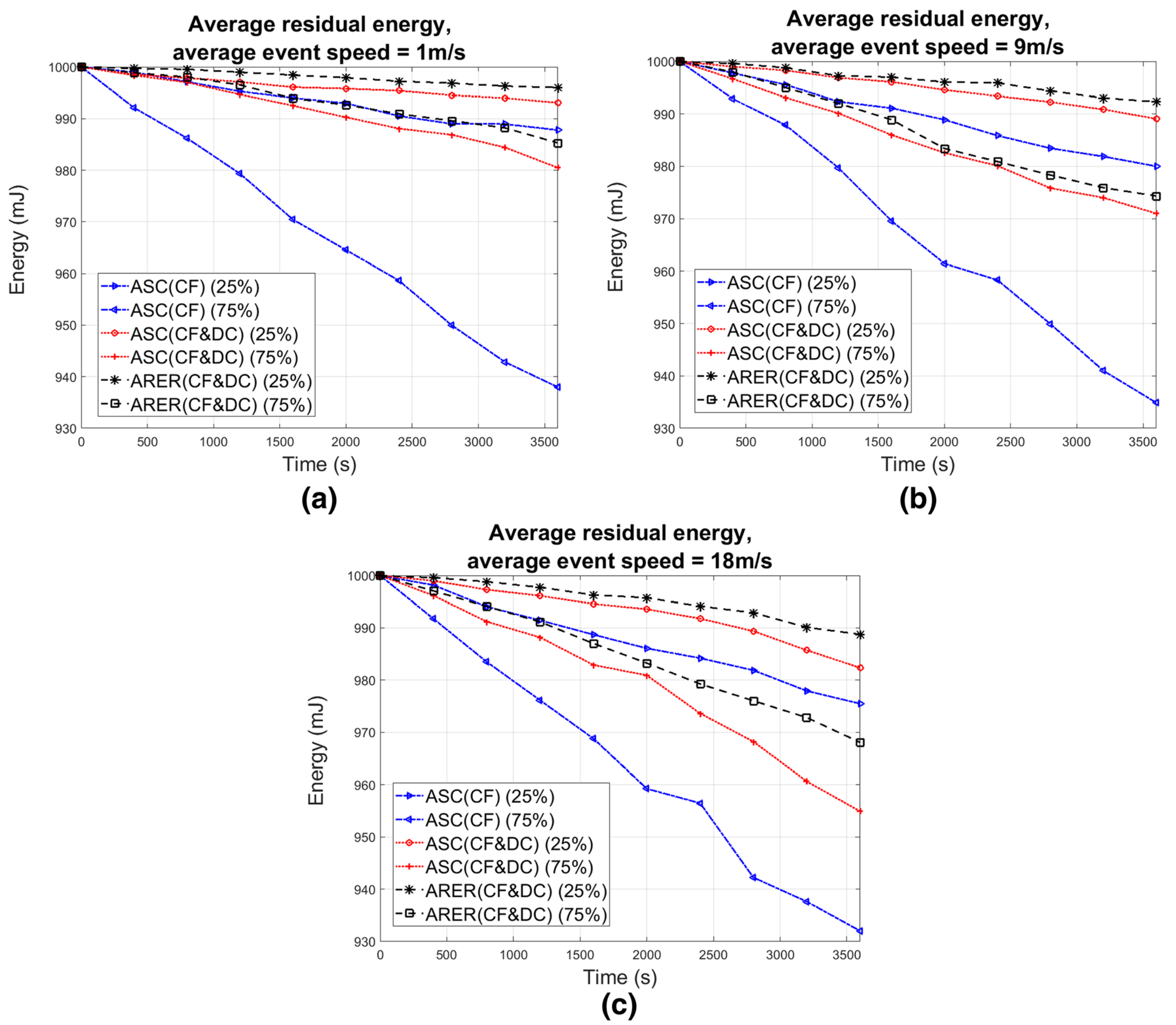

Fig. 4 Average residual energy of the network with different speeds of the event. **a** Average event speed 1 m/s. **b** Average event speed 9 m/s. **c** Average event speed 18 m/s

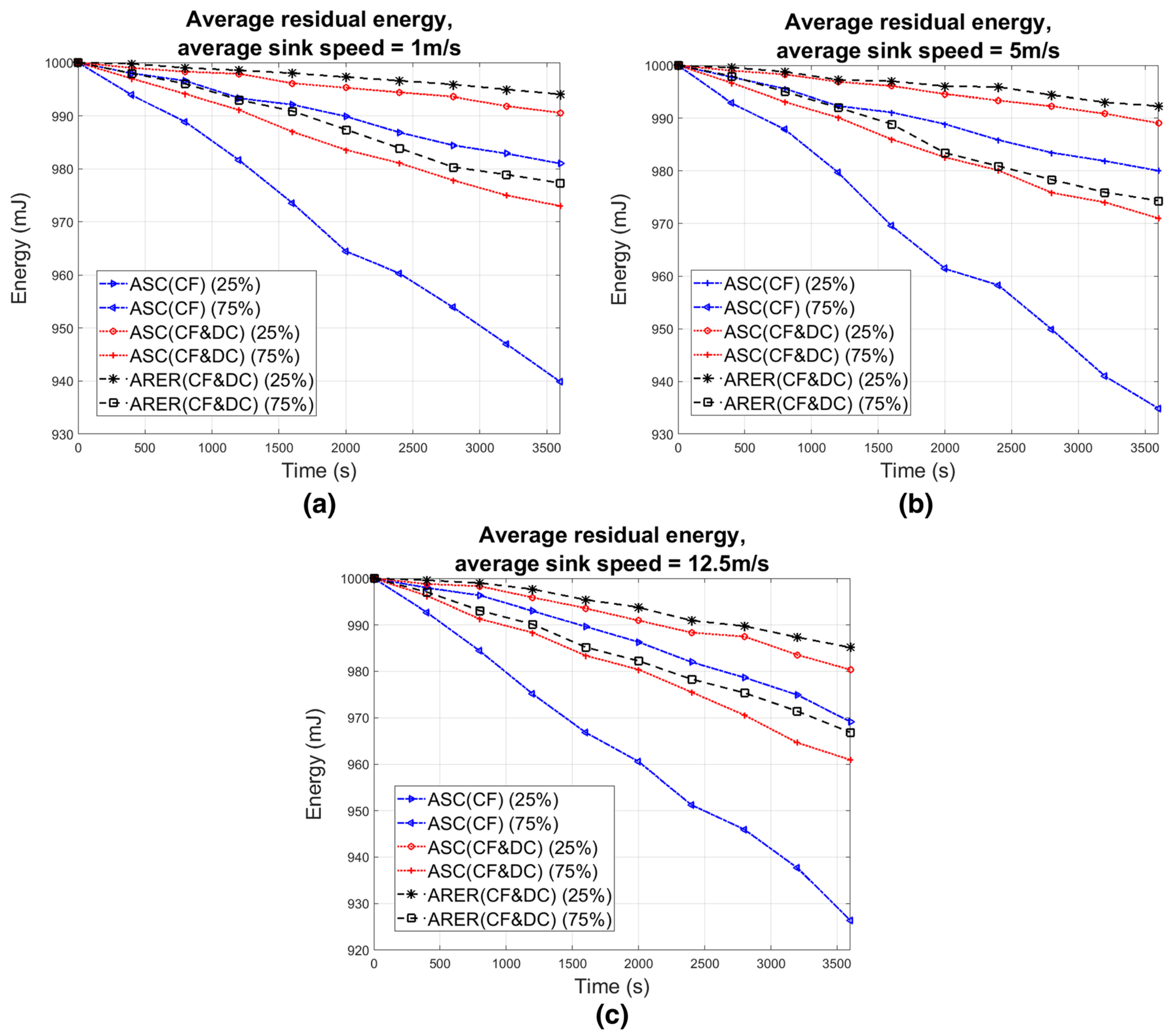

Fig. 5 Average residual energy of the network with different speeds of the sink. **a** Average sink speed 1 m/s. **b** Average sink speed 5 m/s. **c** Average sink speed 12.5 m/s

In Fig. 7, 15 min of simulation time (or 25% of the simulation time) are used to compute the average number of active clusters formed by each algorithm when we vary the event speed, the event size, and the sink speed. The network has $n = 3125$ nodes and $A.L = 1100$ m. The number of clusters vary with the event size. Larger events result in more clusters. A higher event speed results in more clusters. The speed of the event affects the number of clusters because the area of coverage by the event varies, as mentioned previously. The speed of the sink does not affect the number of clusters formed in the network because these values are not dependent. In all measurements, ASC(CF&DC) and ARER(CF&DC) have a smaller number of clusters than ASC(CF) due to the dynamic clustering technique.

For medium events with event speed of 9 m/s, an average of 35.42% of the nodes joining a cluster has changed their cluster at least once. The average number of cluster changes is 1.83, and the maximum number of times a node changes its cluster is 2 (e.g., 3 different cluster memberships during the experiment, as the event moves).

Figure 8 shows the percentage of composite events processed successfully by the sink, when $n = 3125$ nodes and $A.L = 1100$ m. The two experiments vary the average speed of the event and the average speed of the sink, respectively.

We can observe that ARER(CF&DC) improves the percentage of composite events processed successfully by the sink compared to the ASC(CF&DC) protocol. ASC(CF) and ARER(CF&DC) yield similar results, which are better than ASC(CF&DC). ASC(CF&DC) has the smallest

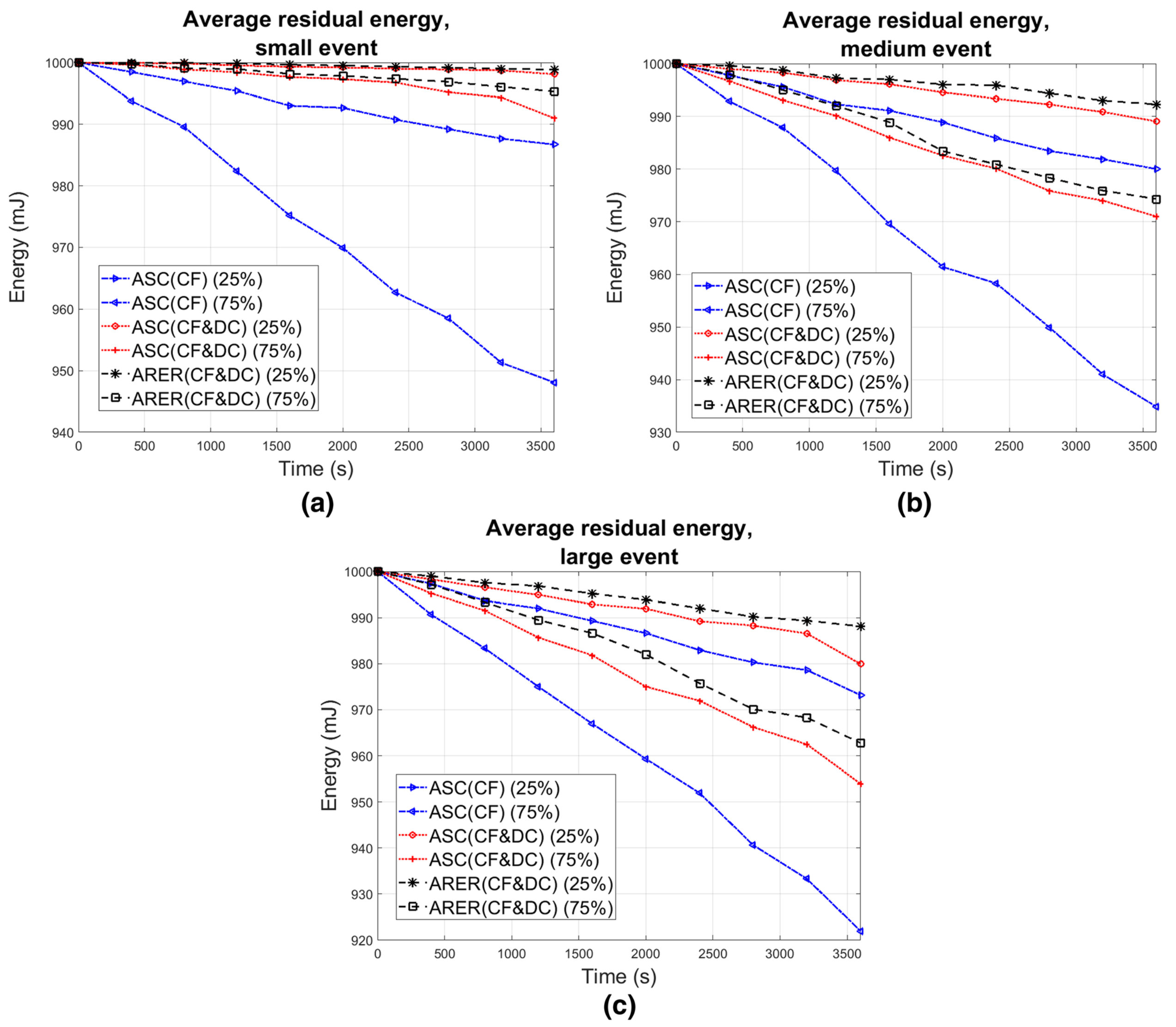

Fig. 6 Average residual energy of the network with different size of the event. **a** Small event. **b** Medium event. **c** Large event

percentage because, when nodes join new clusters (e.g., change clusters), some data packets are lost when the nodes send the acknowledgment to the new cluster head. In the ARER(CF&DC) protocol, no acknowledgment messages are sent when nodes join a cluster.

Both higher sink speed and higher event speed negatively impact the percentage of composite events processed successfully by the sink. A higher sink speed triggers more frequent reconstruction of the convergecast tree, resulting in packets being dropped. A higher event speed leads to new clusters forming more frequently. Packets may be dropped during the cluster formation when messages for joining the cluster and acknowledgment are exchanged, and also some possible contentions and collisions when event-reporting messages are sent to the sink.

From both experiments in Fig. 8, we can see that the percentage of composite events processed successfully at the sink is slightly higher for larger events. This is because more sensors detect the event, thus the redundancy in event reporting helps alleviate the impact of packet dropping.

9 Conclusions

This paper presents ARER, an improvement of the anchor-based routing protocol with constrained flooding and dynamic clustering from [1]. A new clustering algorithm and a new rule for dynamic clustering are proposed. ARER has a better performance in terms of energy consumption and composite events detected successfully by the sink.

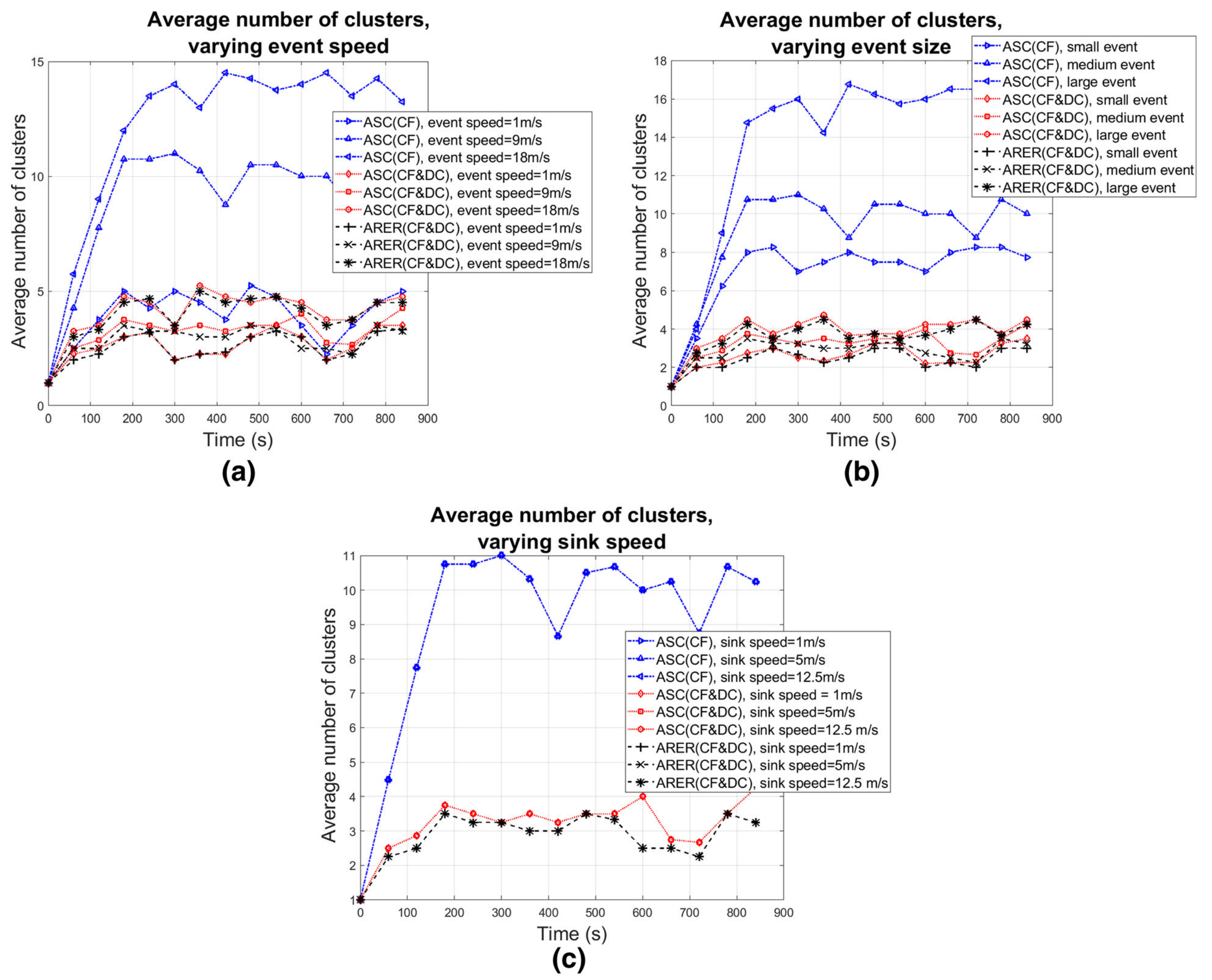

Fig. 7 Average number of clusters in the network. **a** Different event speeds. **b** Different event sizes. **c** Different sink speeds

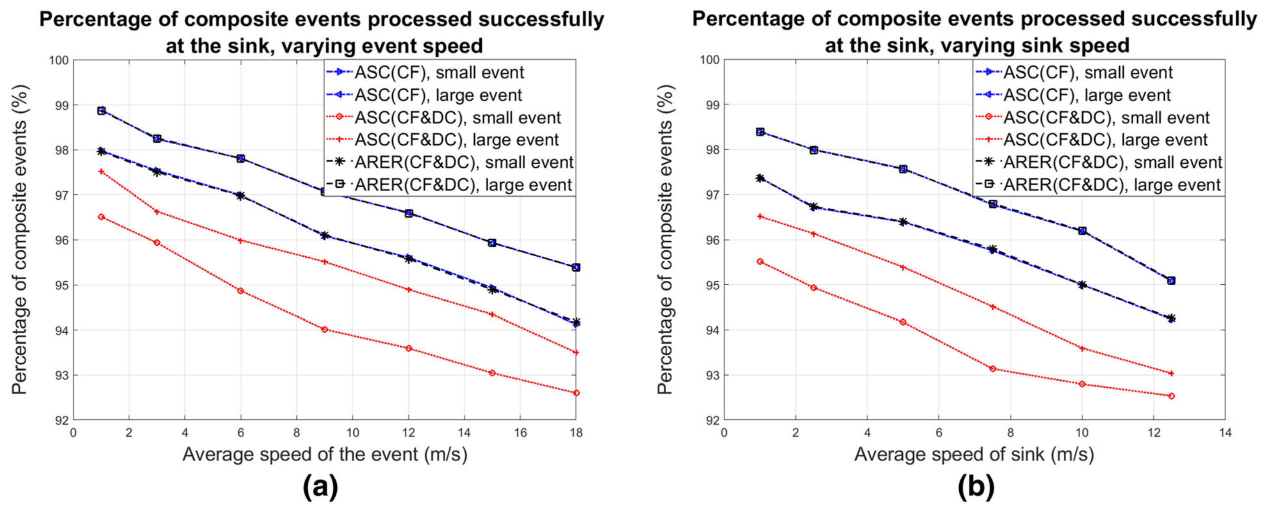

Fig. 8 Percentage of composite events successfully processed at the sink. **a** Different speeds of the event. **b** Different speeds of the sink

Abbreviations

ARER: Anchor-based routing protocol for event reporting.; ASC: Aranzazu-Suescun and Cardei; CF: Constrained flooding; DC: Dynamic clustering; CH: Cluster head; IoT: Internet of Things; STCEDR: Spatio-temporal composite event detection and reporting; WSN: Wireless sensor networks

Funding

This research and the writing of this manuscript has not been supported by any funding.

Availability of data and materials

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Authors' contributions

Both authors have contributed to the design of the routing protocols and planning of the simulations. CAS has conducted the simulation study. Both authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Received: 10 October 2018 Accepted: 25 April 2019
Published online: 23 May 2019

References

1. C. Aranzazu-Suescun and M. Cardei, Spatio-temporal Event Detection and Reporting in Mobile-Sink Wireless Sensors Networks, Proceedings of the 36th International Performance Computing and Communications Conference (IPCCC), 2017
2. C. Aranzazu-Suescun, M. Cardei, Distributed algorithms for event reporting in mobile-sink WSNs for Internet of Things. Tsinghua Sci. Technol. J **22** (2017). https://doi.org/10.23919/TST.2017.7986944C
3. R. Fisher, L. Ledwada, G. Hancke, C. Kruger, Open hardware: A role to play in wireless sensor networks? J. Sens. **15**, 6818 (2015)
4. S. Gao, C.S. Raghavendra, K. Sivalingam, *Composite Event Coverage in Wireless Sensor Networks with Heterogeneous Sensors*, Proceedings of the IEEE Conference on Computer Communications (INFOCOM) (2015), pp. 217–225
5. J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of Things (IoT): A vision, architectural elements and future directions. J. Futur. Gener. Comput. Syst. **29**, 1645 (2013)
6. W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-Efficient Communication Protocol for Wireless Microsensor Networks, Proceedings of the 33rd Hawaii International Conference on System Sciences, 2000
7. F.J. Kaleibar, M. Abbaspour, H.S. Aghdasi, An energy-efficient hybrid routing method for wireless sensor networks with mobile sink. Wirel. Pers. Commun. J. (2016). https://doi.org/10.1007/s11277-016-3434-2
8. A.E. Kostin, Y. Fanaeian, H. Al-Wattar, Anycast tree-based routing in mobile wireless sensor networks with multiple sinks. Wirel. Netw. J. (2016). https://doi.org/10.1007/s11276-015-0975-3
9. S. Kundu and N. Das, Event Boundary Detection and Gathering in Wireless Sensor Networks, Applications and Innovations in Mobile Computing (AIMoC) Conference, 2015
10. S. Kundu, N. Das, S. Roy, D. Saha, *Irregular-Shaped Event Boundary Estimation in Wireless Sensor Networks*, Progress in Intelligent Computing Techniques: Theory, Practice, and Applications, AISC, vol. 719 (Springer, Singapore, 2018), pp. 423–435. https://doi.org/10.1007/978-981-10-3376-6_46
11. Y. Liu, Y. Yang, C. Liu, Y. Gu, Forest fire detection using artificial neural network algorithm implemented in wireless sensor networks. Int. ICT R&D J. **13**, 2 (2015). https://doi.org/10.3969/j.issn.1673-5188.2015.02.003
12. O.F. Mohammed, B. Hussin, A.S.H. Basari, Operational design and modeling of fire event tracking system in wireless sensor networks. ARPN J. Eng. Appl. Sci. **11**, 8525 (2016)
13. D. Morandi, S. Sicari, F. De Pellegrini, I. Chlamtac, Internet of Things: Vision, applications and research challenges. J. Ad Hoc Netw. **10**, 1497 (2012)
14. A. Mousavi, M. Duckham, R. Kotagiri, and A. Rajabifard, Spatio Temporal Event Detection Using Probabilistic Graphical Models (PGMs), IEEE Symposium on Computational Intelligence and Data Mining (CIDM), 2013
15. M. Palaniswami, A.S. Rao, S. Bainbridge, Real-time monitoring of the great barrier reef using internet of things with big data analytics. ITU J. ICT Discov. **1**, 1 (2017)
16. M. Perumal, S. Dhandapani, Modeling and simulation of a novel relay node based secure routing protocol using multiple mobile sink for wireless sensor networks. Sci. World J. (2015). https://doi.org/10.1155/2015/495945
17. P. Rawat, K. Deep Singh, H. Chaouchi, J.M. Bonnin, Wireless sensor networks: A survey on recent developments and potential synergies. J. Supercomput. **68**, 1 (2014)
18. R. Roman, P. Najera, J. Lopez, Securing the Internet of Things. IEEE Comput. **44**, 51–58 (2011). https://doi.org/10.1109/MC.2011.291
19. S.D. Tebje Kelly, N. Kumar Suryadevara, S. Chandra Mukhopadhyay, Towards the implementation of IoT for environmental condition monitoring in homes. IEEE Sensors J. **13**, 3846 (2013)
20. M. Turkanivic, B. Brumen, M. Hlbl, A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion. J. Ad Hoc Netw. **20**, 96 (2014)
21. J. Wang and B. Liu, Online Fault-Tolerant Dynamic Event Region Detection in Sensor Networks via Trust Model, IEEE Wireless Communications and Networking Conference (WCNC), 2017
22. WSNet, an event driven simulator for large scale wireless networks, http://wsnet.gforge.inria.fr. Accessed May 2019
23. J. Yin, D. Hao Hu, and Q. Yang, Spatio-temporal Event Detection Using Dynamic Conditional Random Fields, Proceedings of the 21st International Joint Conference on Artificial Intelligence, 2009
24. D.-G. Zhang, Y.-N. Zhu, C.-P. Zhao, W.-B. Dai, A new construction approach for a weighted topology of wireless sensor networks based on local-world theory for the Internet of Things (IoT). J. Comput. Math. Appl. **64**, 1044 (2012)