

Spatio-Temporal Event Detection and Reporting in Mobile-Sink Wireless Sensors Networks

Catalina Aranzazu-Suescun and Mihaela Cardei

Department of Computer and Electrical Engineering and Computer Science

Florida Atlantic University

Boca Raton, FL 33431, USA

E-mail: {caranzazusue2014, mcardei}@fau.edu

Abstract—Spatio-temporal events are used to model various events such as animal movement, storms (tornadoes, hurricanes, etc.), traffic control, and oil or chemical leakage. Wireless Sensor Networks (WSNs) have been widely used in event monitoring for many applications. Depending on the application, network energy can be a constraint when sensors are battery powered and their batteries cannot be replaced or recharged.

In this paper we propose an anchor-based routing protocol for detecting and reporting spatio-temporal events. Anchor nodes are nodes closer to the sink and they act as relays between the convergecast tree and the sink. The sink is assumed to be mobile. We use several mechanisms to save energy: reactive event reporting, constrained route request flooding with a shortcut mechanism, a shortcut mechanism for data reporting, and dynamic clustering for minimizing the number of active clusters in the network. The performance of our protocol is analyzed using WSN simulator, which is specially designed for event based WSNs. Various metrics such as the average residual energy, the number of active clusters, and the percentage of events processed successfully at the sink are measured.

Index Terms—Wireless sensor networks, mobile sink, spatio-temporal events, anchor based routing, energy efficiency.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) have been widely used in event monitoring applications such as environment, climate, animal monitoring, surveillance and also in the medical field. Events can change in time and space. These spatio-temporal events can change their shape, size, and their movement speed. Spatio-temporal events are used to model dynamic events such as animal movement (e.g. bird migration), storms (e.g. tornadoes, hurricanes), traffic control, and oil or chemical leakage.

Articles [1], [2] present a protocol to detect dynamic events that spread and appear sporadically in an area. When no event is detected, the nodes run the CUSUM (CUmulative SUM) algorithm until an event is detected. The algorithm accumulates log-likelihood ratios of the data collected and if the sum exceeds a threshold, then an event is detected. A node detecting an event runs the P-algorithm (posterior probability distribution algorithm) to detect the spatio-temporal relations between its readings and the readings of its neighbors. When no more events are detected, the node resumes the CUSUM algorithm.

A semantic tree approach [3] is used to detect road traffic. An event detected by a sensor is decomposed into sub-events, following a certain group of semantic rules. The node constructs a semantic tree and a time-series event detection tree that describes the spatio-temporal relation between the event and sub-events. Each neighboring node decides which sub-event to monitor, then it detects and forwards information according with its location and resources.

The event in [4] is disperse and it changes in time and space (i.e. it changes its location). Agents are used in a middleware layer. An agent moves from one sensor to another. A sensor can only communicate with other sensors when it has an agent. The agent checks the readings of the node and if they are greater than zero, then it monitors the area and reports data to the sink. If readings become zero, then the agent checks the readings of its neighbors, and it moves to the neighbor with the highest readings. If no neighbor has readings greater than zero, then the agent is dismissed.

A multi-sink network [5] can be used to reduce congestion due to data reporting. Sinks are placed in the four corners of the area where the network is deployed. Similar to the previous work, the mechanism uses agents. A sensor can be active or inactive. Active nodes send HELLO messages to their neighbors. A node is inactive when no event is detected and no message needs to be forwarded. The agents select coordinator nodes in the network. These coordinators are responsible to route packets to the closest sink.

Spatio-temporal events can be diffuse (e.g. a fire) or moving (e.g. a tornado) [6]. Linear-chain Conditional Random Fields are used. They are undirected graphs that can encode a conditional probability distribution using a specific group of features. They 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 moving.

Article [7] presents a spatio-temporal correlation mechanism used to detect events and report data to the sink. When an event is detected, a cluster is formed and a coordinator (similar to a cluster head) is selected. A correlation zone is created with a set of grids that divide the event zone according to the specifications given by the sink. Each grid has a representative node responsible for sending information to

the coordinator using multihop routing. The coordinator then sends the information to the sink using minimum path. The sink can send a request to re-size the grids of the correlation zone if needed, depending on the type of event detected.

In this paper we propose an anchor-based routing protocol for spatio-temporal event detection and reporting. We assume that such events can move with a certain speed. An important aspect is energy efficiency. We employ several mechanisms to ensure an energy-efficient approach: reactive routing protocol, constrained route request flooding with a shortcut mechanism, a shortcut mechanism for data reporting, and dynamic clustering for minimizing the number of active clusters in the network. More specifically we start with the routing protocol from [13]. Then to better deal with spatio-temporal events and to reduce energy consumption, we design two mechanisms: constrained route request flooding with a shortcut mechanism and dynamic clustering.

The rest of the paper is organized as follows. Section II presents the event model. Section III describes the problem definition. In section IV we present our anchor-based routing protocol for spatio-temporal event detection and reporting. The performance of our algorithm is illustrated in section V, where we conduct simulations using WSNNet [8]. The conclusions are stated in section VI.

II. DESCRIPTION OF THE EVENT MODEL

An event is defined as an observable occurrence of a phenomenon or an object during a period of time in a specific area [9]. We distinguish *atomic events* and *composite events*. An atomic event [9] is triggered when a single sensing value (or attribute) exceeds some threshold and is denoted by $e(t, s, R)$ where t is the time when the event occurs and it can be a specific time or an interval, s is the location of the event and it can be a point or a region, and R is a logical expression defining the conditions when the event occurs. To detect complex events, variations in different attributes have to be detected. A composite event [9] is composed of several atomic events and is denoted by:

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

where e_i , $i=1$ to k , are the atomic events forming the composite event. δ_i with $0 \leq \delta_i \leq 1$ is the confidence of e_i , indicating the probability of E occurring when e_i occurs. R_i is a logical expression defining when e_i occurs. C_t is the constraint on atomic events' times t_1, t_2, \dots, t_k . C_s is the constraint on atomic events' locations s_1, s_2, \dots, s_k . The confidence δ of the composite event is defined as $\delta = \delta_1 + \delta_2 + \dots + \delta_k$, and it is expected to satisfy the property $\delta_1 + \delta_2 + \dots + \delta_k = 1$ [9].

We consider spatio-temporal events similar to [6]. An event moves with a certain speed, but we assume that it maintains the same shape, see Figure 1. More specifically, we assume that events have circular shape and move on a random path.

III. PROBLEM DEFINITION

We consider a WSN consisting of n heterogeneous nodes N_1, N_2, \dots, N_n and a mobile sink S . The nodes are densely

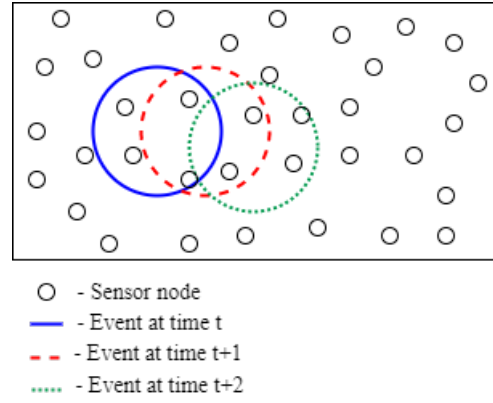


Fig. 1: Spatio-temporal event

deployed, they have the same communication range R_c and the same initial energy E_{init} . The sink S has communication range R_c , infinite energy, and is mobile. Table I shows the main notations in the paper.

Each node is equipped with one or multiple sensing components from the set $\{s_1, s_2, \dots, s_m\}$. Each sensing component can be used to detect an atomic event for that attribute. Nodes may have different sensing components [10], [11]:

- nodes may have been manufactured with different sensing capabilities
- some nodes may have purposely turned off some sensing components to save energy
- some sensing components may fail over time
- some sensing components cannot be used due to lack of memory for storing data.

Nodes in WSNs are resource constrained in terms of power, bandwidth, memory, and computing capabilities. Since replacing or recharging nodes' battery is often impractical or infeasible, the mechanisms designed for event detection and reporting have to minimize the energy consumption in order to prolong network lifetime [10], [11]. We assume that the nodes are synchronized since the deployment, so we do not deal with node synchronization in this paper.

Depending on the application, the events can change their shape, can appear/disappear (e.g. fire event), and they can move (e.g. storms such as hurricanes, tornadoes). These events are called spatio-temporal events [6]. In this paper we assume that events have a circular shape and they are mobile, following a random path. Next, we present the problem definition.

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 $\{s_1, s_2, \dots, s_m\}$ and a mobile sink S , design an energy-efficient 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 $\{s_1, s_2, \dots, s_m\}$.

TABLE I: 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 , $1 \leq j \leq n$
$N_j.\{s_{j_1}, s_{j_2}, \dots, s_{j_k}\}$	Sensing components of node N_j , $1 \leq k \leq m$
$N_j.E_{residual}$	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 $T_{cluster}$
R_c	Node communication range
A	Deployment area
$A.L$	Length of the side of the area
E_{init}	Initial energy of each node

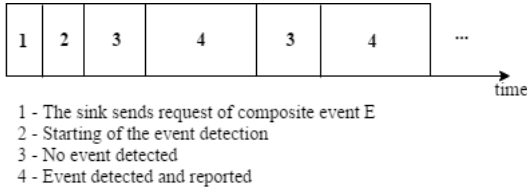


Fig. 2: Network organization

IV. ANCHOR-BASED ROUTING PROTOCOL FOR STCEDR IN MOBILE-SINK WSNS

In this section we propose an anchor-based routing protocol for STCEDR in mobile-sink WSNs. We start with the anchor-based routing protocol [13] which we describe in section IV-A. This protocol is further improved to reduce energy consumption and to better deal with spatio-temporal events. To improve energy-efficiency, we propose a constrained route-request flooding with a shortcut mechanism, presented in section IV-B. For spatio-temporal events, the clusters of nodes detecting the event change over time. To reduce the number of active clusters in the network, and thus to reduce energy spent on event reporting, we propose a dynamic clustering mechanism in section IV-C.

A. Anchor-based Routing Protocol

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 floods the request for monitoring the composite event E , through A_1 . More specifically, S sends *SinkInitiatedRequest*(S, A_1, E, δ_{th}), where E is the composite event and δ_{th} is the threshold parameter for the composite event.

A_1 then broadcasts *CompositeEventRequest*(A_1, E, δ_{th} , $hops = 0$) in the whole network. 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 for the first

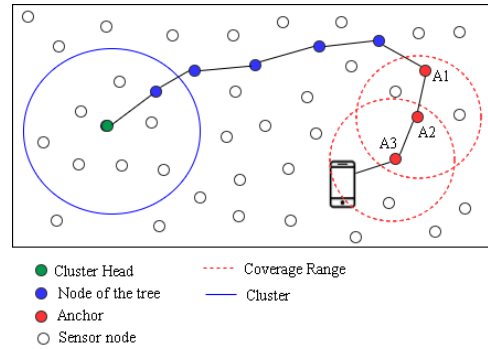


Fig. 3: New anchor selection

time, increments the *hops* field, sets the sending node as its parent in T , stored in the field $N_j.tp$, and sends a message *CompositeEventRequest*($N_j, E, \delta_{th}, hops$). At the end of this step, each node N_j has set-up its parent in the tree T in the field $N_j.tp$.

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, \dots, e_k , start the detection process for a time duration C_t .

In *phase 3*, no atomic event part of the composite event requested by the sink S is detected. This phase takes zero or more time.

In *phase 4*, one or more nodes start detecting the event and initiate the mechanism for event reporting. One or more clusters are formed using the *event-based clustering* algorithm from [10]. The cluster contains nodes that detect atomic events part of the composite event requested by the sink and relay nodes to connect the sensing nodes to the CH. 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 the residual energy and *ID* (used to break ties), a node proclaims itself CH and sends a message *JoinCluster* over $h_{cluster}$ hops. The nodes in the cluster form a cluster tree $T_{cluster}$ rooted at the CH. $T_{cluster}$ is expected to have a small height $h_{cluster}$, such as 2 or 3. Since there is no guarantee that all the nodes detecting atomic events are within $h_{cluster}$ -hops of the CH, additional clusters may form. A CH receives atomic events from cluster members, which are sent along $T_{cluster}$. As message are sent from cluster members to the CH, aggregation is performed.

At the end of *phase 2*, each node has a pointer to its parent in the convergecast tree T rooted at A_1 . Event report messages flow from the nodes to the CH, from CH to A_1 along T , and from A_1 to S . Since S is mobile, we need a mechanism to deal with the case when S moves out of A_1 's range. A_1 sends beacons (or data) periodically. If S does not hear a beacon (or data) from A_1 for α periods (e.g. $\alpha = 2$), then S selects a new anchor A_2 as follows. S broadcasts a message *NewAnchorRequest*(S, A_1). Nodes that receive both A_1 's beacons (or data) and S 's message are candidates to become A_2 , since they are connected to both A_1 and S . Such a node N_j waits a time based on

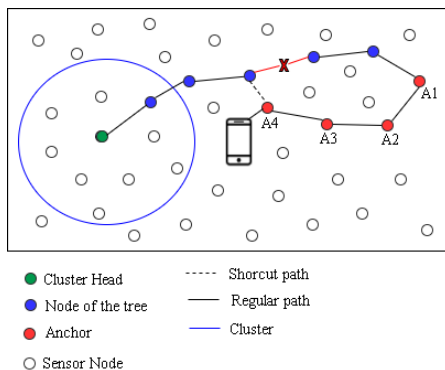


Fig. 4: Example of the *shortcut* mechanism

the signal strength of the message *NewAnchorRequest*, and sends a message *NewAnchorReply*(S, A_1, N_j). The waiting time is smaller when the signal strength is higher. When the first message is received by the sink, S replies with *NewAnchorAck*(S, A_1, N_j), and N_j 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 Figure 3. After the maximum number of anchors β is reached, the anchor selection process resets, that means a new anchor A_1 is selected.

Events may move or even 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 considered obsolete and removed. Depending on the length of the previous *phase 3*, the fields may be deleted or not. If the parent fields are not obsolete, then data flows from CH to A_1 along T , and from there to S along the chain of anchors.

If the parent fields are obsolete, then the CH has to find a path to reach the sink S . The CH broadcasts *RouteRequest*(S). When S receives the message, it selects A_1 using the mechanism described previously and then A_1 sends back in the whole network a *RouteReply* with the parameters of the composite event request. We use flooding to send the reply rather than the sole path to the CH since often times more clusters are formed and in this way we avoid another *RouteRequest* being initiated by other CHs.

After the convergecast tree T is formed, the event is reported from the CH to the anchor A_1 along the tree T using the parent attribute tp . From A_1 the event is reported directly to the sink (if A_1 is the last anchor) or is using a path of at most β anchors. The attribute ap stores the next anchor in the path to the sink. For example, for $\beta = 3$, $A_1.ap = A_2$, $A_2.ap = A_3$, and $A_3.ap = S$.

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 is sending data directly to the anchor instead of sending it through the rest of the path. This mechanism is illustrated in Figure 4.

Phases 3 and *4* can interleave, see Figure 2. In *phase 3*,

if no data is transmitted for the duration γ , then the parent fields, the cluster information, and the anchor path fields are obsolete.

B. Anchor-based Routing Protocol with Constrained Flooding

In *phase 3*, if a new CH has its parent field obsolete, it has to find a path to S to report its data. Rather than flooding the *RouteRequest*(S) in the whole network, we use an incremental ring search mechanism. The message is sent with a certain TTL h_{req} which indicates the number of hops used.

Note that some nodes might have the tp field active, while others may have this field obsolete. We implement a shortcut mechanism. If any node with active tp field receives a *RouteRequest* message, then it replies with *RouteReply* which contains the number of hops to A_1 or the number of hops to the last anchor if beacons are received. If the CH receives more *RouteReply* messages, then it sets-up a path to the node with a shortest-path to S . The nodes along that path set-up their tp field accordingly.

If no *RouteReply* message is received, then h_{req} is increased according to the incremental ring search mechanism, and in the end the whole network is flooded.

C. Anchor-based Routing Protocol with Constrained Flooding and Dynamic Clustering

Spatio-temporal events may move in time. Usually one or more 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 events. The nodes which do not detect the event any longer will not transmit data, and as a result parts of a cluster may become obsolete and the parent field times-out. A cluster is active as long as some data is reported to the CH and from there to the sink.

As new nodes start detecting the event, new clusters are formed. When a new cluster is formed, the CH sends a *JoinCluster* message, as described previously in *phase 4*. The nodes that are not part of a cluster, detect atomic events, and receive the *JoinCluster* message join the new cluster by sending a reply to the CH.

We propose a *dynamic clustering* mechanism. A node N_j that is already part of a cluster and receives a *JoinCluster* message from another CH, joins the newer cluster. Actually if more *JoinCluster* messages are received, then N_j joins the newest cluster. This mechanism can be implemented by having the CH add a time-stamp field in the *JoinCluster* message.

The advantage of using the dynamic clustering is that as the spatio-temporal event moves, older clusters become obsolete faster, thus fewer clusters are active and transmit messages to the sink. This mechanism is expected to reduce the number of event reporting messages from CHs to the sink.

V. SIMULATIONS

A. Simulation Environment

We conduct simulations using WSNNet [8]. The main parameters used in simulation are listed in Tables II,III,IV,V. WSN

TABLE II: Simulation parameters

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

TABLE III: 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

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

$n=3125$ nodes are randomly deployed in the area. The maximum number of sensing components is $m=5$. Each node is equipped randomly with sensing components. We define a composite event with five atomic events. The sensing components involved, with confidence and threshold values are shown in Table III. The five atomic events are:

- $e_1(t_s, A, temperature > 150)$
- $e_2(t_s, A, pressure > 50)$
- $e_3(t_s, A, humidity > 10)$
- $e_4(t_s, A, smoke > 100)$
- $e_5(t_s, A, light > 80)$.

t_s is the simulation time after the request is sent by the sink S and A is the deployment area. The composite event is $E((e_1, 0.35), (e_2, 0.1), (e_3, 0.15), (e_4, 0.3), (e_5, 0.1), t_s, A, \delta)$.

The initial energy of each node is $E_{init}=1Joule$. We use the energy model from LEACH [14] to measure the energy consumption:

$$E_{Tx}(l, d) = E_{elec} * l + \epsilon_{amp} * l * d^2$$

$$E_{Rx}(l) = E_{elec} * l$$

where $E_{elec} = 50nJ/bit$, $\epsilon_{amp} = 100pJ/bit/m^2$ and d is the distance between nodes. In our simulation, a node is a candidate to become a CH only if its residual energy is at least $900mJoules$.

In each simulation run, we generate an event which has a circular area. The center is generated randomly. Three types of events are used in the simulations:

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

The nodes located in the event area, equipped with the corresponding sensing components, detect an atomic event with probability 95%. Simulation time is $1hr$. A node detecting the event sends report messages to the sink S every 5s.

The sink moves in the area A using a random walk mobility model with the average and maximum speeds indicated in

TABLE IV: Sink speed

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

TABLE V: Event speed

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

Table IV. The sink pauses for some time, then it moves with a speed between 0 and the maximum value for a random time. The direction angle has a random value between 0° and 360° .

The spatio-temporal event has a random move, using a random walk mobility model with the average and maximum speeds indicated in Table V.

B. Simulation Results

In this section we compare three algorithms:

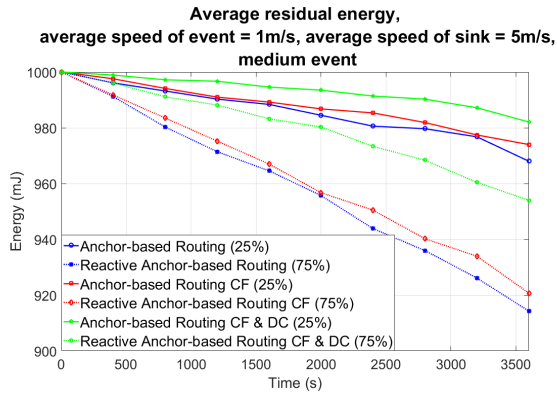
- Anchor-based routing protocol from section IV-A, called *Anchor-based routing*
- Anchor-based routing protocol with constrained flooding from section IV-B, called *Anchor-based routing CF*
- Anchor-based routing protocol with constrained flooding and dynamic clustering from section IV-C, called *Anchor-based routing CF & DC*.

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

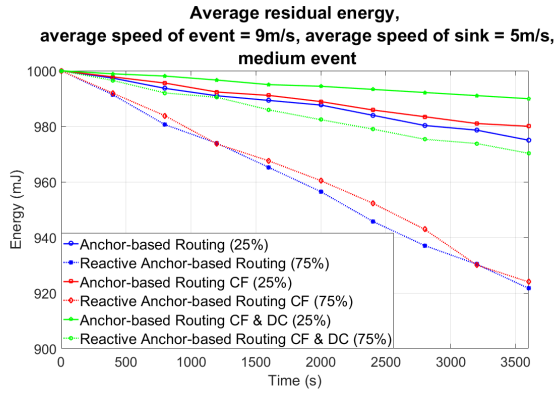
In all experiments, Anchor-based routing CF & DC has the least energy consumption. This is due to the dynamic clustering mechanism which accelerates the timeout of the older clusters. Thus fewer clusters are involved in data reporting. The Anchor-based routing protocol consumes the most energy since it uses flooding to establish a path to the sink. The constrained flooding technique with the shortcut mechanism has some improvements in energy consumption due to the incremental ring search and the shortcut mechanisms.

More clusters are formed when the event has a smaller speed. As the event slowly moves, a smaller new area is covered by the event. New sensors start detecting the event and new clusters are formed. Therefore many smaller clusters are formed. On the other hand, if the event moves faster, fewer larger clusters are formed and the old clusters timeout sooner.

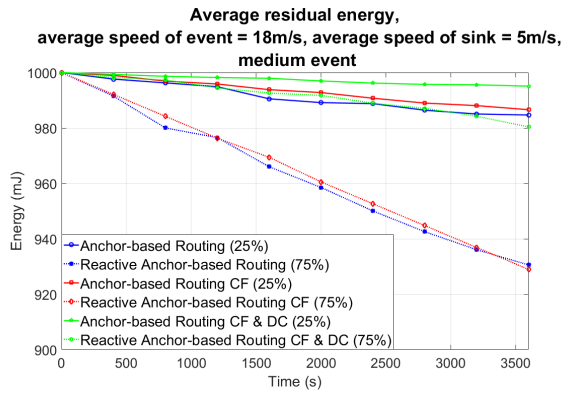
Figure 6 shows the residual energy of the network for $n = 3125$ nodes, $A.L = 1100m$, medium size events, and average event speed $9m/s$. The average sink speed varies between $1m/s$, $5m/s$ and $12.5m/s$. When the sink moves faster, new anchors have to be selected and when the maximum



(a)



(b)

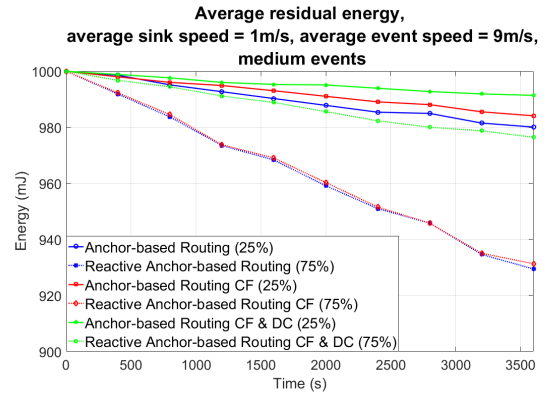


(c)

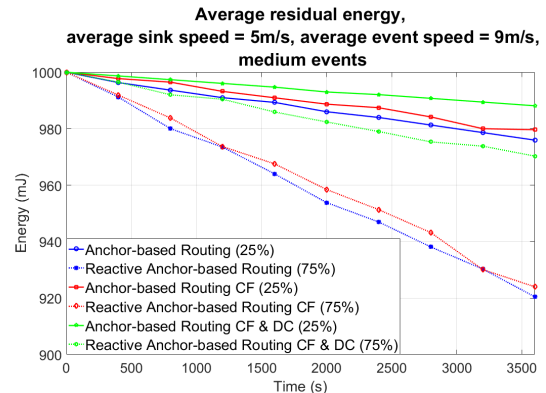
Fig. 5: Average residual energy of the network with different speeds of the event. (a) Average event speed $1m/s$. (b) Average event speed $9m/s$. (c) Average event speed $18m/s$.

number of anchors is reached, a new convergecast tree is built. More energy is therefore spent on this process. Similar to the previous experiment, Anchor-based Routing consumes the most energy, followed by Anchor-based Routing CF and Anchor-based Routing CF & DC. The highest reduction in energy is due to the dynamic clustering mechanism.

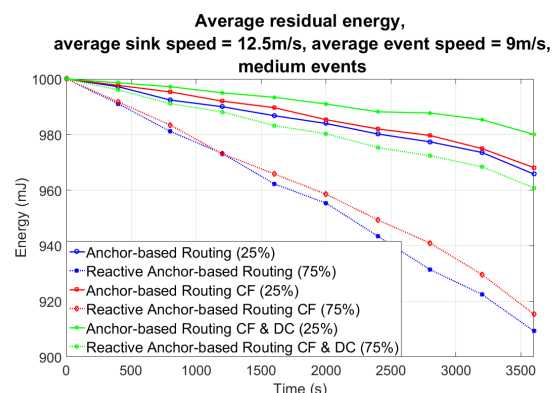
In Figure 7, the average sink speed is $5m/s$, the average event speed is $9m/s$, and the number of nodes is $n = 3125$. Results are measured for small, medium, and large events with



(a)



(b)

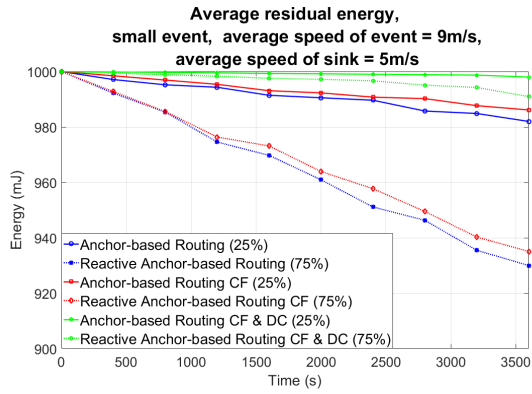


(c)

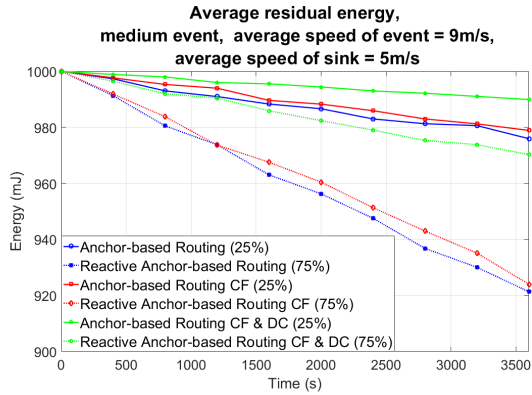
Fig. 6: Average residual energy of the network with different speeds of the sink. (a) Average sink speed $1m/s$. (b) Average sink speed $5m/s$. (c) Average sink speed $12.5m/s$.

one event in the network. More energy is spent by the network on data reporting for larger events. This is because more nodes detect the event and participate in data reporting. Also, more clusters are formed.

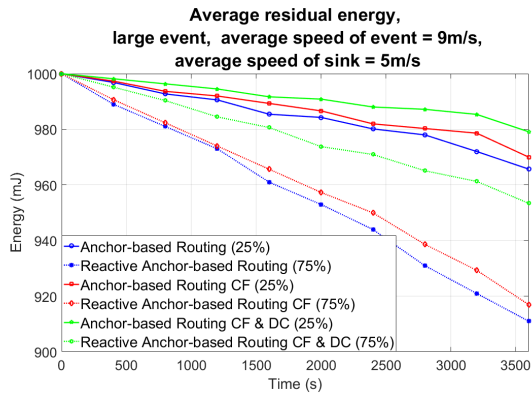
Figure 8 shows the average number of clusters formed by each algorithm when we vary the event speed, the event size, and the sink speed. The network has $n = 3125$ nodes, $A.L = 1100m$, and the event duration is $15min$. The number



(a)



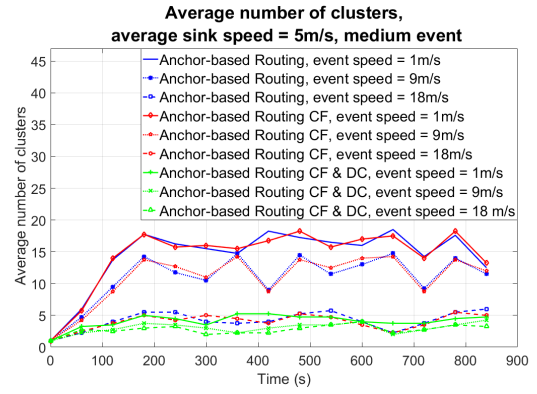
(b)



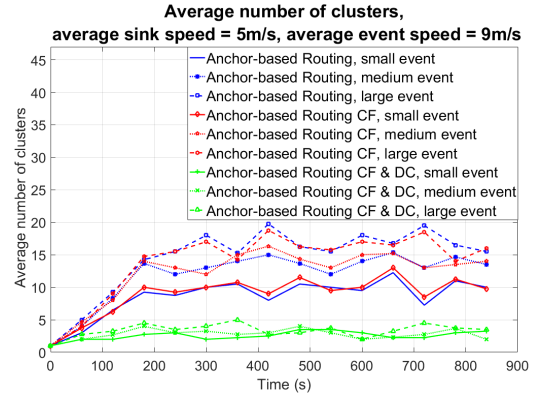
(c)

Fig. 7: Average residual energy of the network with different size of the event. (a) Small event. (b) Medium event. (c) Large event.

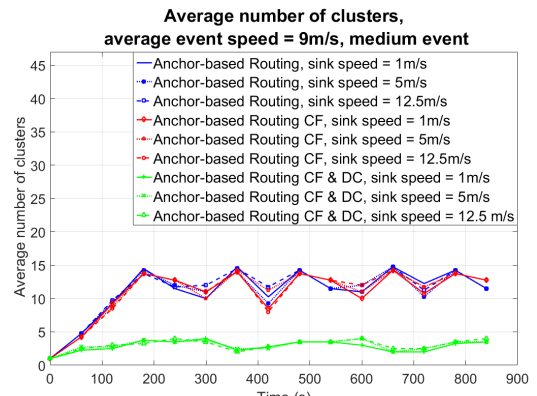
of clusters vary with the event size; larger events are more likely to have more clusters. Also, the speed of the event affects the number of clusters because the area of coverage by the event varies. As argued previously, a smaller event speed leads to more clusters in average. Contrary, the speed of the sink does not affect the number of clusters formed in the network, because these values are not dependent. In all measurements, the Anchor-based Routing CD & DC has the least number of clusters due to the dynamic clustering



(a)



(b)



(c)

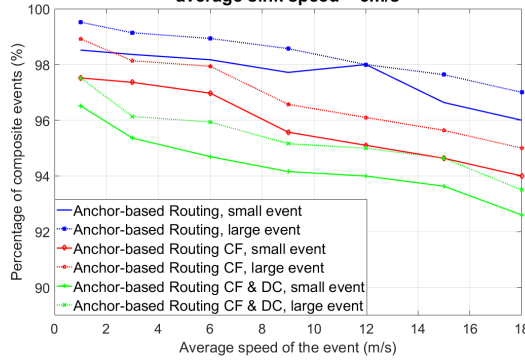
Fig. 8: Average number of clusters in the network. (a) Different event speeds. (b) Different event sizes. (c) Different sink speeds.

mechanism.

Figure 9 presents the percentage of composite events successfully processed by the sink in the network for $n = 3125$ nodes and $A.L = 1100m$. The two experiments vary the average speed of the event and the average speed of the sink, respectively.

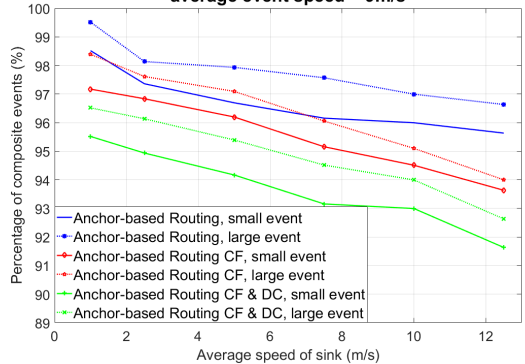
Overall, Anchor-based Routing has the best results, followed by Anchor-based Routing CF and Anchor-based Routing CF & DC. The algorithm with constrained flooding

Percentage of composite events processed successfully at the sink,
average sink speed = 5m/s



(a)

Percentage of composite events processed successfully at the sink,
average event speed = 9m/s



(b)

Fig. 9: Percentage of composite events successfully processed at the sink. (a) Different speeds of the event. (b) Different speeds of the sink.

approach has a poorer outcome since more packets are dropped when the incremental ring search mechanism is run. Sometimes few iterations are run, resulting in more packets dropped. The shortcut mechanism may result in more packets sharing the same path, leading to packet lost due to contentions and collisions.

The algorithm with constrained flooding and dynamic clustering has the smallest percentage of composite events processed successfully at the sink, but always greater than 90%. In dynamic clustering, nodes join the new clusters (e.g. change clusters), and in this process some packets are lost while the nodes send the acknowledgment and new clusters are formed.

Both higher sink speed and higher event speed negatively impact the percentage of composite events processed successfully at the sink. A higher sink speed triggers a 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 *JoinPackets* and acknowledgment packets are exchanged.

From both experiments in Figure 9, 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.

VI. CONCLUSIONS

This paper presents an energy-efficient anchor-based routing protocol for detecting and reporting spatio-temporal events to the sink. The mechanism uses two novel techniques: constrained flooding with a shortcut mechanism and dynamic clustering. Simulation results show that these techniques reduce the overall energy consumed by the network. The Anchor-based Routing CF & DC has the best energy performance and the least number of clusters. The percentage of composite events processed successfully by the sink is above 90% and it can be further improved using caching during the execution of the incremental ring search mechanism.

REFERENCES

- [1] H. J. Enemark, Y. Zhang, N. Dragoni and C. Orfanidis, Energy-efficient fault-tolerant dynamic event region detection in wireless sensor networks, *IEEE 81st Vehicular Technology Conference (VTC Spring)*, DOI: 10.1109/VTCspring.2015.7145814, 2015.
- [2] T. Wu and Q. Cheng, Online dynamic event region detection using distributed sensor networks, *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 1, pp. 393405, DOI: 10.1109/TAES.2013.120308, Jan. 2014.
- [3] H. Wu, J. Cao, and X. Fan, Dynamic collaborative in-network event detection in wireless sensor networks, *Telecommunication Systems*, DOI: 10.1007/s11235-015-9981-0, Mar. 2016.
- [4] J. L. Fernandez-Marquez, J. L. Arcos, and G. Di Marzo Serugendo, A decentralized approach for detecting dynamically changing diffuse events in noisy WSN environments, *Applied Artificial Intelligence*, pp. 376-397, DOI: 10.1080/08839514.2012.653659, 2012.
- [5] S. Kamal and P. Varalakshmi, Energy efficient and congestion avoidance event tracking in wireless sensor networks, *International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN)*, DOI: 10.1109/ICSCCN.2011.6024536, Sep. 2011.
- [6] J. Yin, D. Hao Hu, and Q. Yang, Spatio-temporal event detection using dynamic conditional random fields, *21st International Joint Conference on Artificial Intelligence*, Jul. 2009.
- [7] L. A. Villas, A. Boukerche, D. L. Guidoni, H. de Oliveira, R. Borges de Araujo, and A. Loureiro, An energy-aware spatio-temporal correlation mechanism to perform efficient data collection in wireless sensor networks, *Computer Communications Journal*, DOI: 10.1016/j.comcom.2012.04.007, Apr. 2012.
- [8] WSNeT - An event driven Simulator for large scale Wireless Networks (online): <http://wsnet.gforge.inria.fr/>, last accessed Jul. 2017.
- [9] J. Gao, J. Li, Z. Cai, and H. Gao, Composite event coverage in Wireless Sensor Networks with heterogeneous sensors, *INFOCOM 2015*.
- [10] C. Aranzazu-Suescun and M. Cardei, Event-based clustering for composite event detection in wireless sensor networks, *IEEE International Performance Computing and Communications Conference (IPCCC)*, pp.366-373, Dec. 2016.
- [11] M. Marta, Y. Yang, and M. Cardei, Energy-efficient composite event detection in wireless sensor networks, *WASA 2009*.
- [12] C. Aranzazu-Suescun and M. Cardei, Distributed algorithms for event reporting in mobile-sink WSNs for Internet of Things, *Tsinghua Science and Technology Journal*, Vol. 22, Aug. 2017.
- [13] C. Aranzazu-Suescun and M. Cardei, Reactive routing protocols for event reporting in mobile-sink wireless sensor networks, *The 20th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, accepted for publication.
- [14] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, *33rd Hawaii International Conference on System Sciences*, Jan. 2000.