



**HAL**  
open science

## Towards a Smarter Data dissemination strategy in connected environments

Sabri Allani, Richard Chbeir, Khouloud Salameh, Taoufik Yeferny

### ► To cite this version:

Sabri Allani, Richard Chbeir, Khouloud Salameh, Taoufik Yeferny. Towards a Smarter Data dissemination strategy in connected environments. *Procedia Computer Science*, 2020, 176, pp.1043-1052. 10.1016/j.procs.2020.09.100 . hal-03010037

**HAL Id: hal-03010037**

**<https://hal.science/hal-03010037>**

Submitted on 17 Oct 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License



24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

## Towards a Smarter Data dissemination strategy in connected environments

Sabri Allani<sup>a,\*</sup>, Richard Chbeir<sup>a</sup>, Khoulood Salameh<sup>b</sup>, Taoufik Yeferny<sup>c</sup>

<sup>a</sup>Univ. Pau & Pays Adour, UPPA E2S, LIUPPA, Anglet, France

<sup>b</sup>American University of Ras Al Khaimah, Ras Al Khaimah, UAE

<sup>c</sup>LIPAH-LR 11ES14, University of Tunis El Manar, 2092 Tunis

---

### Abstract

Since the last decade, the emergence of affordable smart devices in web-connected environments (e.g., smart cities and buildings) has been a critical step towards improving user comfort. In this respect, notifying end-users about different essential events in their environment is of significant interest. Thus, the design of an efficient data dissemination strategy is of paramount importance. The thriving challenge would be to inform interested devices about an event and to avoid as far as possible the network collision problem. To overcome these limitations, we introduce a new generic and adaptive data Dissemination Strategy in Connected Environments (DSCE). Our strategy's main thrust is reaching a high delivery ratio as well as a low network overhead by sending messages only to the interested devices. To do so, we rely on a semantic-based approach to define device interests and to identify salient messages. Moreover, to cope with the broadcast storm problem and to organize the scheduling of data dissemination, we provide a new waiting time measure for each device before rebroadcasting any received message. The performed experiments show that our strategy outperforms its competitors in terms of effectiveness and efficiency.

© 2020 The Author(s). Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Peer-review under responsibility of KES International.

*Keywords:* Connected Environment, Internet of Things, Data dissemination, Broadcasting.

---

### 1. Introduction

Over the last decades, the growth of connected devices (particularly smartphones) has expanded enormously, which gave birth to various smart fields, such as smart cities, smart buildings, smart health, and smart home [1]. In order to provide end-users useful monitoring and analysis means, devices need to exchange various messages using three

---

*E-mail address:*

sabri.allani@univ-pau.fr\*

richard.chbeir@univ-pau.fr

khoulood.salameh@aurak.ac.ae

yeferny.taoufik@gmail.com

1877-0509 © 2020 The Author(s). Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Peer-review under responsibility of KES International.

© 2020 published by Elsevier. This manuscript is made available under the CC BY NC user license

<https://creativecommons.org/licenses/by-nc/4.0/>

communication ways: Device-to-Device (D2D) communication [2], where devices communicate directly. Device-to-Infrastructure (D2I) communication [3], where devices communicate through infrastructure such as a Road Side Unit (RSU). Hybrid Communication (D2X), where devices can combine both techniques D2D and D2I to communicate. However, these underlying communication channels need to cope with: 1) various types of events to be detected and related messages, and 2) each device's preferences to send only appropriate notifications [4]. For instance, some devices would be interested in non-safety events (e.g., commercial advertisement, publicity, social information, ...). In contrast, others would be concerned about safety events (e.g., accidents, security-related, environment warnings, ...). Therefore, a core and challenging issue in connected environments is the design of a smart data dissemination strategy to exchange information between all devices based on each event type and its priority [5]. Interestingly enough, a successful data dissemination approach needs to cope with the following requirements:

- Deliver the message only to interested devices
- Inform devices about interesting and critical events with the minimal delay
- Maintaining important notifications alive as long as possible while filtering out the number of inappropriate messages so to avoid broadcast storm problem [6].

To cope with the challenges mentioned above, we propose here a new generic data Dissemination Strategy for Connected Environments (DSCE). Our approach aims to ensure a high reachability ratio as well as a high dissemination precision by sending data only to the interesting devices with a minimum overhead cost. The principal novelty of strategy is identifying the interested devices through the unveiling of strong associations between devices' properties and events' categories. To do so, we rely on a semantic model to associate interesting events with relevant devices.

The rest of this paper is organized as follows. Section 2 outlines the current approaches and their disadvantages. Section 3 presents a description of our dissemination strategy. In Section 4, we present the experiments carried out to validate our approach, and the key results obtained. Section 5 concludes the paper and sets out several future directions.

## 2. Related work

In this section, we detail the main existing approaches of the data dissemination strategies in connected environments. Those strategies are split into three categories: Device to Device (D2D), Device to Infrastructure (D2I), and Hybrid (D2X) communications.

### 2.1. Device to Device Communication (D2D)

The D2D architecture is based on an inter-devices communication that does not involve any infrastructures. In this type of communication, a connected environment is considered as a particular case of MANET [7].

In [8], the authors proposed a framework for handling local broadcast storms using probabilistic data dissemination and aggregation in a connected vehicular environment. The dissemination process is based on multi-criteria decision-making for data management. The proposed approach allowed the devices to determine the relevance of gathered information (e.g., accidents, network collisions, broken devices, ...). This later is determined by calculating the semantic and syntactic similarities of two or more inputs, allowing to eliminate the duplication and to decrease the overhead. To this end, the authors represented the gathered information as a KD-tree data structure to verify the relevance of graph characteristics between nodes. Although this solution can decrease the overhead ratio and avoid the broadcast storm problem [9], it can't guarantee a high dissemination precision.

### 2.2. Device Communications to Infrastructure (D2I)

In the D2I architecture, the devices can communicate through infrastructures, such as Road Side Units, satellites, routers, switchers, IoT clouds, antennas, internet, data centers or any other service points [1].

In [10], the authors proposed a D2I dissemination algorithm based on what they called "cluster-Head2Devices" and "Device2cluster-Head" dissemination, which takes into account three parameters: The energy quantity of each

device The position of the device to the other members of the cluster The distance between the cluster members and the cluster sink Those mentioned above prioritize the device having the highest energy, near to the cluster-head. Note that the closest device to the remote sink is chosen to be a cluster-head. Consequently, the power usage of the network is reduced, and the life cycle of the connected environment network is enhanced. Although, this dissemination method ensures a high reachability ratio, it also causes high latency and a low precision ratio due to the uninterested devices notification.

In [11], the authors proposed a data dissemination and power conservation in connected environments, based on cluster heads and members. The proposed method adopted the modified butterfly optimization approach [12] to form the cluster. The load distribution model promotes associative clustering depending on the remaining energy and overhead factors of the devices in the cluster. Added to that, members of the cluster disseminate the data to the cluster head through multi-hop communication with load balancing methods. The distribution of loads in the clusters prevented energy consumption. Therefore, the efficient routing took place from clusters to access points utilizing the multi-hop technology [13]. In this respect, the interested clusters in an event were based only on the remaining devices' energy, which could decrease the precision of the dissemination. For instance, the uninterested devices with the highest left energy will receive a lot of messages. In contrast, other interested devices will not receive this message, or they received it after a high latency delay.

### 2.3. Hybrid Communications (D2X)

The hybrid communication (D2X) is the combination of both architectures D2D and D2I [2]. The use of devices as a relay of communication allows not only extending the restricted ranges of the infrastructure but also preserving its usage. A particular case of the D2X is Wireless Sensor Networks (WSN)[14].

In [15], the authors presented a routing protocol for Adhoc networks called: AODV (Ad hoc On-Demand Distance Vector). This protocol is capable of both multicast and unicast routing. It is free of loops, self-starts, and serves a vast number of mobile or intermittent nodes. Besides, for multicast groups, AODV builds a tree structure. AODV protocol is energy-efficient and does not require a lot of computing power, making it convenient to be installed on small mobile equipment. However, when a connection breaks down, many control messages are created, which increases the CE's overhead level. Further, AODV needs high latency time and high processing demand to build its routing table.

### 2.4. Discussion

To sum up, the existing dissemination schemes have several limitations, but mainly:

- Send messages only to the interested devices.
- Avoid the broadcast storm during the dissemination of events to devices within the region of interest by using a moderated Slotted-1 persistence technique [6]. In fact, this suppression technique avoids the broadcast storm problem and guarantees a high *reachability*.
- Smart switching between the D2I and D2D dissemination techniques.

## 3. Proposed Data Dissemination Strategy for Connected Environments (DSCE)

Before describing our strategy, we introduce in the following subsection some related definitions and concepts.

### 3.1. Preliminaries

A Smart Connected environment (CE) is a set of connected devices, which are capable of collecting and transmitting data independently in a limited geographical area in order to provide end-users with appropriate monitoring and analysis. Devices can be static or mobile within the CE area. The main parts of the CE are defined in our approach as follows:

**Definition 1. Device (D) :** A device "D" in a CE can be any object (e.g., mobile sensor, vehicle, drone, smart phone, ...). A device can detect and receive messages. In our approach, we assume that all devices are able to process (receive, send) received messages while some are dedicated to the detection of events. It has the following attributes:

- **Id:** represents the device identifier;
- **MetaData:** characterise a device with:  $\langle \text{Location, Description, Detection\_Features} \rangle$ , where "Location" is the spatial location of the device, "Description" contains a description that can be provided by the admin of CE, and "Detection features" enumerates the types of events that can be detected;
- **Type:** represents the device type: static or mobile
- **Beacon frequency:** represents a fixed delay in seconds to determine the periodic dissemination interval;
- **Connected To Infrastructure:** represents a Boolean value that indicates whether a device is connected to an infrastructure or not.
- **Device range:** is distance value that represents the communication range of the device.
- **Interests:** represents a set of values that defines the category of interest (e.g., Temperature, Weather, Fire).
- **List\_of\_Events:** represents the set of events detected by the device.

**Definition 2. Message (M):** Devices of the CE exchange messages with the followings properties:

- **Id:** represents the message identifier;
- **Sender:** represents the last device that sends the message M;
- **Type:** represents the message type (e.g., Safety, Sport, Break);
- **Data:** represents the disseminated information. It can be an event or raw data (e.g., video, audio, scalar value);
- **Priority\_Time:** represents a waiting delay that a device should wait before disseminating the message M;
- **Life\_time:** represents the expiration delay, after which the message is no longer be useful;
- **Labels:** represents a list of values that defines the categories of the message data (e.g., weather, Temperature, Humidity). This list of labels will be generated by the event classifier (cf. Device architecture in Figure 1).

To keep the packet size of our message small, the maximum allowed size of the message is set according to 802.11p standard [16].

**Definition 3. Event (E):** is anything of interest that happens or occurs in a CE. The event structure is defined as follows:

- **Id:** represents the event identifier;
- **Event\_Space :** is the target region of interest of the event (e.g., GPS location, room number, street);
- **Creator\_device:** represents the first device that detects the event;
- **Description:** represents a value that describes the event. It is generated by the event classifier.

### 3.2. Device Architecture at a Glance

As depicted in Figure 1, the device architecture includes the following components:

- **Event detector:** receives signals from device's sensors then it determines the event location. In addition, it allows to search for events received by neighbouring devices located in the communication range of current device.
- **Event receiver:** receives messages from other devices, organises and plans the scheduling of the events.
- **Decision Maker:** decides whether the device is interested or not in an event message. However, the decision maker calculates the similarity between the messages based on the cosine similarity matching between their labels. Thus, it decides if a device D is interested to a message m.
- **Event Classifier:** classifies events and predicts their life time and dissemination priority. Basically, it identifies the main category of the event (e.g., a safety or non-safety event), and then determines its priority level (e.g., high or low priority). To this end, the event classifier was trained using a local database (where previous events

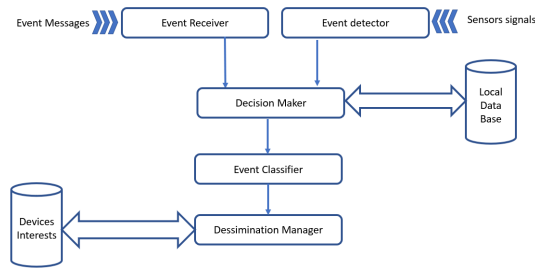


Fig. 1. Device Architecture

have been stored) to classify new events and assign to them the relevant labels. This component won't be described further in this paper and will be detailed in a dedicated study.

- **Device interests:** is a database containing event categories that have some interests to the device (e.g., safety, non-safety, road events, sports publicity).
- **Local Database:** is used by each device to store the previous detected or received events.
- **Dissemination Manager:** disseminates events to other devices in the CE through wired or wireless networks. In some situations, when the device cannot connect to an existing infrastructure, wireless ad-hoc communications between devices is used to disseminate events.

In the following subsection, we thoroughly describe the different mechanisms used in our DSCE strategy to disseminate data in various scenarios.

### 3.3. The DSCE Strategy

The flowchart diagram in Figure 2 depicts the different operations performed by each device running our DSCE strategy to handle detected or received events. In what follows, we detail the main processes executed whenever the device detects a new event or receives an event message from other devices.

#### 3.3.1. Event Detection Process

Upon detecting an Event  $E$ , the event detector creates a new message  $m$ , which be transferred to the decision maker component. This latter checks whether the message  $m$  is similar or not to the previous received or detected messages. If there is a high similarity between  $m$  and the previous seen messages then  $m$  will be discarded. Otherwise, the decision maker adds  $m$  to the local database then sends it to the event classifier component. In this stage, the event priority is computed and its life time is predicted. Thereafter, the message  $m$  will be disseminated to the interested devices via the dissemination manager component. Two key operations are executed by this component to ensure an efficient and effective network communications. These operations are explained latter in the subsection 3.3.3.

#### 3.3.2. Event Reception Process

Whenever the event receiver of the device  $D$  receives an event message  $m$ , it automatically forwards it to the decision maker component, which will check whether  $m$  is an interesting message to  $D$ . In this case, it computes the similarity between  $m$  and the previous seen messages. If the similarity value is low, the decision maker will add  $m$  to the local database then submits it to the event classifier component. Otherwise, the message  $m$  will be ignored. Furthermore, if  $D$  is not interested by  $m$ , it automatically transmits it to the event classifier. At this step, the event life time and priority are estimated then the message will be sent to dissemination manager.

#### 3.3.3. Dissemination Mechanism

One of the main goals of our DSCE strategy is to have a smart switching between D2D and D2I communications. In fact, static device broadcasts the message according to a beacon frequency defined by the admin of the CE. While mobile device must wait for a certain delay (i.e., Priority Waiting Time) before broadcasting the message in order to avoid the broadcast storm problem. After the waiting time period, the message will be only disseminated to the

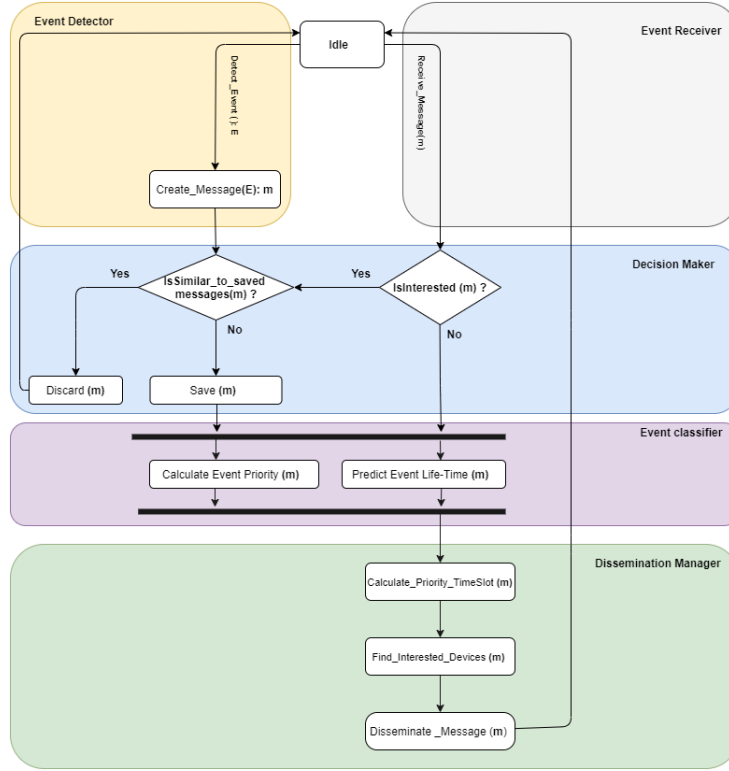


Fig. 2. Flowchart Diagram of the Proposed Data Dissemination Strategy

interested devices. In the following, we detail how the priority waiting time is computed and how we determine the interested devices to a given event message.

#### • Priority Waiting Time calculation:

As we mentioned in the related work Section, one of our challenges is to reduce the overhead ratio and to avoid the broadcast storm problem [6]. To this end, we propose an optimized version of the waiting time-slotted method adopted in [6]. The later is based only on the device range and the distance between devices to calculate the waiting time. However, this method [6] is adapted only to mobile devices and a for a static event life time. In this respect, our main optimization is to include the message life time  $F(m)$  and the message priority  $P(m)$ . This consists of calculating a priority waiting time slot  $T_{m,xy}$ , once a device receives a message. Thus, the device will re-disseminate the message if it does not receive any redundant message during the waiting priority time; otherwise, it withdraws it. Given all of the relative distances  $D_{xy}$  between the devices  $x$  and  $y$ , and the average transmission range  $R$ , the waiting time  $T_{m,xy}$  is computed as follows:

$$T_{m,xy} = (1 - [\frac{\min(D_{xy}, R)}{R}]) \times F(m) \times P(m) \quad (1)$$

where  $F(m)$  is the estimated message life time and  $P(m)$  is the message priority. Those two values are defined by the event classifier. This time slot calculation process is depicted in Algorithm 1. Figure 3 illustrates the process of the priority time distribution between devices by showing an example of a priority time distribution between five devices, where the smallest waiting time  $T(m)$  is given to the farthest device.

Doing this, the device avoids the storage of duplicated messages and reduces the overhead ratio. Consequently, we avoid the dissemination storm problem [6, 17].

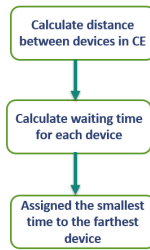


Fig. 3. Priority time slot calculation

---

**Algorithm 1** Algorithm of priority waiting time calculation

---

```

1: function CALCULATE_PRIORITY_TIMESLOT(Message M)
2:    $d \leftarrow \text{Distance}(m.sender.MetaData.location, This.MetaData.location)$ 
3:    $r \leftarrow This.Range$ 
4:    $Mf = m.lifetime$ 
5:    $Mp = m.priority$ 
6:   if (This.Type == "Static") then
7:     return This.Beacon.Frequency
8:   else
9:      $TimeSlot = Mf * Mp (1 - (\min(d,r)/range))$ 
10:    return TimeSlot
11:  end if
12: end function
  
```

---

- **Finding interested devices**

After calculating the waiting priority time of a message  $m$ , the dissemination manager determines through algorithm 2 the list of interested devices of that message.

---

**Algorithm 2** Find interested devices for a message  $m$

---

```

1: List LD  $\leftarrow$  List_of_Connected_Devices
2: List F  $\leftarrow$  Null
3: function FIND_INTERESTED_DEVICES(Message m)
4:   for (Int I = 0 ; I  $\leq$  LD.Length ; I++) do
5:     if (LD[I].IsInterested(m.Labels)) then
6:       F  $\leftarrow$  LD[I]
7:     end if
8:   end for
9:   Return F
10: end function
  
```

---

Algorithm 3 depicts the data dissemination mechanism. In fact, before sending a message  $m$ , the device needs to find the list of other interested devices using Algorithm 2. Then, it calculates the waiting time of this message, and disseminates it through one of the following communication methods:

- SendD2D: allows to disseminate the message to the interested devices without involving infrastructures.



- SendD2X: allows to make a hybrid communication between other devices and infrastructures. Via this method, a device D can use other devices as a relay of communication to extend the restricted range.

---

**Algorithm 3** Dissemination process
 

---

```

1: procedure SEND_MESSAGE (Message m)
2:    $T \leftarrow \text{This.CalculatePriorityTimeSlot}(m)$ 
3:    $F \leftarrow \text{This.Find_Interestred\_Devices}()$ 
4:   Wait ( $T$ )
5:   if ( $\text{This.Connected\_to\_Infrastructure}()$ ) then
6:     This.SendD2X ( $M, T, F$ )
7:   else
8:     This.SendD2D ( $M, T, F$ )
9:   end if
10: end procedure

```

---

## 4. Experimental Evaluation

In this section, we present the performance evaluation of our proposed DSCE versus: 1) the conventional broadcasting technique [3] and 2) the enhanced Ad-hoc On-demand Distance Vector (AODV) multicast protocol [15]. We chose the AODV protocol as a baseline approach because it is capable of both unicast and multicast routing. AODV is loop-free, self-starting, and accommodates a large number of static and mobile devices.

To conduct our experiments, we simulated a WSN connected environment using the network simulator OMNeT++ 5.5.1 [18]<sup>1</sup> along with the physical layer modelling toolkit INET<sup>2</sup>, which allows to accurate models for radio interference. We used the same simulation settings of AODV defined in [15].

For an impartial analysis, we simulated the AODV protocol and our proposed approach using similar metrics as illustrated in Section 4.2.

### 4.1. Experimental scenarios

In our tests, we varied from twenty to one hundred connected devices. In order to cope with our aforementioned objectives mentioned in Section 1,

### 4.2. Evaluation metrics

The assessment of the performances of our solution is carried out through the following metrics:

**Overload:** stands for the total number of sent messages. Interestingly enough, the ultimate goal of any dissemination is to avoid the overload problem [2] by looking for minimizing the number of messages exchanged in the network.

**Reachability:** assesses the average delivery ratio of dissemination, where the message must reach all intersected devices of such an event message  $m$ . The reachability is defined as follows:

$$\text{Reachability}(m) = \frac{\text{Informed Devices}(m)}{\text{Interested Devices}(m)} \quad (2)$$

<sup>1</sup> <https://omnetpp.org/>

<sup>2</sup> <https://inet.omnetpp.org/>

where the set of interested informed devices i.e., only pertinent devices for an event message  $m$ , and the set of interested devices in an event message  $m$ .

**Precision:** evaluates if dissemination strategy is able to only inform devices that are interested in a given message  $m$ . The metric of precision is defined formally as follows:

$$Precision = \frac{|IIV|}{|AIV|} \quad (3)$$

where  $IIV$  stands for the set of interested informed devices, i.e. only pertinent devices for message  $m$ , and  $AIV$  stands for the set of all informed devices, i.e. pertinent as well as not pertinent devices for a message  $m$ . The average precision is defined as follows:

**Latency:** refers to the amount of time needed to deliver a message to an interested device.

#### 4.3. Results

As expected, the reachability of the broadcasting is very high (cf. Figure 4(c)). This is due to informing all devices (i.e., interested or not). Also, one can notice that in existing approaches, the values of the reachability and precision decrease as far as the number of devices increases. Indeed, the higher the number of devices, the lower the probability of reaching the interested devices. Added to that, we observe that the reachability of the AODV approach is slightly sharper than our strategy 4(c). This is due, within the AODV strategy, to the fact that the message is sent nearly to all the devices in the network (i.e., interested or not) since the circular region with a range equal to 300m will include the whole area in most cases. Hence, a higher number of devices is notified, increasing the reachability metric, consequently. However, our strategy ensures a very high reachability ratio that is around 90%.

Figure 4(d) demonstrates that our strategy has a high dissemination precision under a different number of connected devices since it only targets interested devices and keeps a low overhead value. Whenever compared to the AODV or Broadcast, our strategy performs a better tradeoff reachability/precision than the baseline protocol.

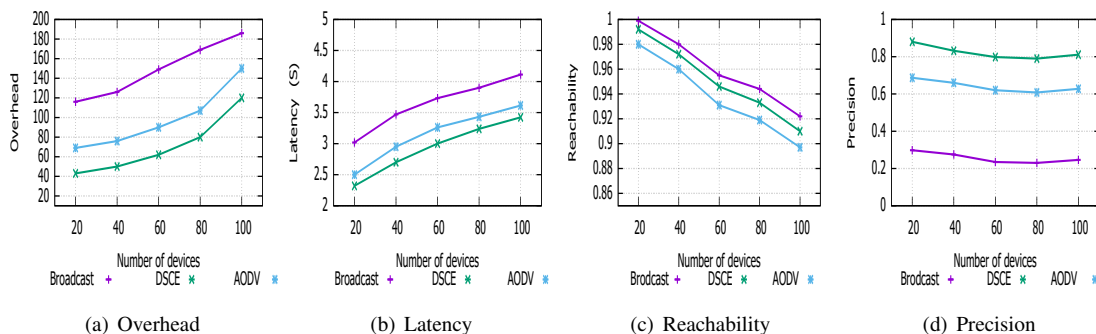


Fig. 4. Variation of the average Overhead/Latency values w.r.t the variation of the number of devices

Figure 4(b) shows that our approach outcomes the lowest latency compared to the other approaches. The reason behind is that our protocol flags out shorter notification delay and minimum network overhead; this increases the overload level, which consequently increases the network collisions and negatively affects the latency [2].

As expected, using our dissemination strategy, the overhead level is decreased (cf. Figure 4(a)). This happens thanks to the use of our optimized version of the priority waiting time slotted approach[6] and the high precision value by targeting only the interested devices.

To sum up, simulations demonstrated that our proposed solution outperforms the baseline strategies from various perspectives such as overhead, latency, and dissemination precision. Also, our new strategy is generic and capable of being adapted for any CE.

## 5. Conclusion and future work

In this paper, we have introduced a generic and adaptive solution for data dissemination in a CE context. Compared to the existing approaches, our strategy overcomes several limitations, such as broadcast storm problem, high overhead level, and network effectiveness. Simultaneously, our aforementioned objectives have been achieved, namely: i) disseminating data with the minimum overhead ratio and maximum precision for each device, ii) Smart data dissemination based on the event priority, and iii) Inform devices about critical events with the minimal delay. Besides, we have demonstrated through experimental simulations that our solution has good optimized results compared to the one provided by the baseline approach [15, 3]. Avenues of future work are as follows:

- Working on real-world validation and planning scenarios
- Proposing a privacy-based framework for our strategy.

## References

- [1] S. Allani, R. Chbeir, K. Salameh, Towards better data management/gathering in connected environments, in: *Proceedings of the 15th ACM International Symposium on QoS and Security for Wireless and Mobile Networks, Q2SWinet'19*, Association for Computing Machinery, New York, NY, USA, 2019, p. 53–59.
- [2] S. Allani, R. Chbeir, T. Yeferny, S. Ben Yahia, Smart directional data aggregation in VANETs, in: *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*, Vol. 2018-May, 2018, pp. 63–70.
- [3] V. Sharma, A. Vij, Broadcasting methods in mobile ad-hoc networks, *Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2017 2017-January (2017)* 582–587.
- [4] T. Yeferny, S. Allani, Mpc: A rsus deployment strategy for vanet, *International Journal of Communication Systems* 31 (12) (2018) 1–18.
- [5] S. Allani, T. Yeferny, R. Chbeir, S. B. Yahia, Towards a smarter directional data aggregation in VANETs, *World Wide Web* (2020).
- [6] N. Wisitpongphan, O. Tonguz, J. Parikh, P. Mudalige, F. Bai, V. Sadekar, Broadcast storm mitigation techniques in vehicular ad hoc networks, *Wireless Communications, IEEE* 14 (6) (2007) 84–94.
- [7] S. Varshney, R. Kuma, Variants of leach routing protocol in wsn: A comparative analysis, in: *2018 8th International Conference on Cloud Computing, Data Science Engineering (Confluence)*, Vol. 1, 2018, pp. 199–204.
- [8] R. Kumar, M. Dave, A framework for handling local broadcast storm using probabilistic data aggregation in vanet, *Wireless Personal Communications* 72 (1) (2013) 315–341.
- [9] S. Allani, T. Yeferny, R. Chbeir, A scalable data dissemination protocol based on vehicles trajectories analysis, *Ad Hoc Networks* 71 (2017) 31–44.
- [10] Y. C. Wang, G. W. Chen, Efficient data gathering and estimation for metropolitan air quality monitoring by using vehicular sensor networks, *IEEE Transactions on Vehicular Technology* 66 (8) (2017) 7234–7248.
- [11] D. Sundaranarayana, K. Venkatachalapathy, Energy preservation in mobile ad-hoc networks using a modified butterfly optimization with associative cluster head load distribution, in: *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, Vol. 2, 2018, pp. 448–453.
- [12] C. F. Olson, Parallel algorithms for hierarchical clustering, *Parallel Computing* 21 (8) (1995) 1313–1325.
- [13] A. Sumalee, H. W. Ho, Smarter and more connected: Future intelligent transportation system, *IATSS Research* 42 (2) (2018) 67–71.
- [14] I. Ud Din, M. Guizani, S. Hassan, B. S. Kim, M. Khurram Khan, M. Atiqzaman, S. H. Ahmed, The internet of things: A review of enabled technologies and future challenges, *IEEE Access* 7 (2019) 7606–7640.
- [15] A. Singh, T. Singh, M. Mittal, K. Kumar, Performance improvement in aodv routing protocol with artificial intelligence, in: *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2016, pp. 45–49.
- [16] F. C. Commission, Amendments of the commission's rules regarding dedicated short-range communication services in the 5.9 ghz band, *Federal Communications Commission Proceedings* (2004) 10–809.
- [17] I. D. Chakeres, E. M. Belding-Royer, Aodv routing protocol implementation design, in: *24th International Conference on Distributed Computing Systems Workshops, 2004. Proceedings.*, 2004, pp. 698–703.
- [18] V. Andras, Omnet++, in: *Wehrle, Klaus, Gones, Mesut, Gross, James (Eds.), Modeling and Tools for Network Simulation*, 2010, pp. 35–59.