Ad Hoc Networks 47 (2016) 53–81

Contents lists available at ScienceDirect

Ad Hoc Networks

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

A survey on clustering techniques for cooperative wireless networks

Victor Sucasas[a,b,*], Ayman Radwan[a], Hugo Marques[a,b], Jonathan Rodriguez[a], Seiamak Vahid[b], Rahim Tafazolli[b]

[a] Instituto de Telecomunicações, Aveiro, Portugal
[b] Institute for Communication Systems, University of Surrey, UK

ARTICLE INFO

Article history:
Received 9 July 2015
Revised 22 April 2016
Accepted 23 April 2016
Available online 3 May 2016

Keywords:
Clustering
Ad hoc networks
Cooperative networks

ABSTRACT

Clustering became relevant in the past as a solution for the scalability problems of ad hoc networking, but, the unsuccessful application of ad hoc solutions to real scenarios, such as the projects SURAN and PRNet, decreased the interest of research community on ad hoc communications, and subsequently, on clustering algorithms. Recently, however, clustering techniques have gained renewed interest due to the emergence of cooperative communications for cellular networking. Clustering is envisaged, in this scenario, as a technique to team up nodes to support efficient data aggregation for energy saving, scalability and privacy among other benefits. Moreover, research on 5G networks also envisages a connected society, where everything and everyone will be connected under the umbrella of Internet of Everything (IoE). This novel communication paradigm has fostered new research on clustering, which has yielded novel and more advanced algorithms and applications. This article surveys the State-of-the-Art in clustering techniques and provides detailed descriptions of the basics of clustering and the latest novel ideas. Open issues, technical challenges and directions for future research are also outlined.

1. Introduction

Clustering, in telecommunication systems, appeared in the framework of ad hoc networking, which was proposed decades ago as a promising solution for infrastructure deprived networks, such as military battlefield communications, emergency communication black-outs or to deploy fit for purpose wireless networks [1]. The idea of setting a network, on-demand, without implementing access points or base stations, was very appealing from both the utility and economic perspective. In ad hoc networking, the nodes communicate among themselves by transmitting and relaying the packets from source to destination, without a common hub that routes the packets or organizes the communication flow. This paradigm, however, requires complex routing mechanisms to optimize the route path selection and ensure a satisfactory packet delivery ratio. This was indeed the bottleneck of ad hoc networking, since the unpredictable nature of this scenario, where nodes show up and vanish continuously, requires a continuous update of the routing information. This information can grow significantly in big networks, due to the countless route paths that routing algorithms must evaluate. Clustering was suggested in this framework to reduce the routing complexity and provide scalability. The strategy of clustering is teaming up nodes in virtual groups, with one leader per group, such that routing techniques consider these groups as sole entities, hence reducing the number of network sources, destinations and possible paths considerably, hence increasing its stability.

The research interest in clustering techniques went side by side with the concern of ad hoc networking, very prolific during two decades, producing research efforts in a variety of scenarios and technologies such as mobile wireless networks and sensor networks. This interest however has been gradually declining due to the absence of implementation of ad hoc solutions in real scenarios. With the sole exception of wireless sensor networks, ad hoc networking has been considered a liability problem, and the main projects on ad hoc solutions for military and emergency situations have failed [2,3]. Clustering, originally designed for ad hoc networking, lost some relevance for research community. However, the emergence of more advanced wireless devices such as smartphones, wearable technology and vehicular networks, provided with higher processing capabilities and multi-standard wireless interfaces, has catered a new scenario for clustering in the framework of cellular networks and cooperative communications. In this scenario, cooperation provides many benefits and only one requirement. The benefits include energy efficiency, scalability, efficient location services, social networking and privacy preserving among others. The requirement is an efficient technique to establish cooperative groups of nodes, i.e. clusters.

* Corresponding author at: Instituto de Telecomunicações, Aveiro, Portugal.
E-mail address: vsucasas@av.it.pt (V. Sucasas).

http://dx.doi.org/10.1016/j.adhoc.2016.04.008
1570-8705/

Energy saving can be accomplished by using different wireless standards for the short range and long range links, i.e. a low energy demanding wireless interface, such as Bluetooth, can be used in short range to aggregate traffic, whereas a higher range and higher energy demanding link will be used to relay this traffic. Clustering is in this scenario envisaged to provide a stable cooperative framework between nodes [4]. Multi-standard wireless transmission is not the only way to achieve energy efficiency, nodes can also transmit information, in short range links with high transmission bit rates, to the relay nodes with better channel conditions, hence with the capability to transmit at high order modulation for longer periods, which effectively reduces the transmission time [5]. This strategy has been proposed in different scenarios, including wireless sensor networks [6] or mobile wireless networks [7]. Moreover, efficient location services [8] and mobile social networking [9] for MANETs can be supported by cooperative clustering.

Vehicular networking also considers the approach of teaming up nodes for scalability [10] and privacy reasons [11]. Traffic monitoring applications require a continuous transmission of information updates from vehicles to a backend server, hence this information goes through the base station. According to current analysis of the state of the art, this continuous transmission of information can form a bottleneck in the LTE uplink connection. Data aggregation has been proposed in this scenario to reduce the number of uplink connections [10]. In short range, vehicles can transmit and aggregate traffic using the Dedicated Short Range Connectivity (DSRC, 802.11p) wireless interface, then the relay nodes transmit the aggregated traffic over the LTE uplink. Regarding privacy preserving for vehicular networks, it can also be provided by aggregating nodes into clusters, since data aggregation hides the identities of the nodes forming the cluster (only the identity of the leader of the cluster is revealed) [11]. Internet connectivity is also envisaged as a desirable commodity for VANETs, where again QoS in terms of jitter, delay, packet delivery ratio or connection duration are the most relevant metrics. These requirements have inspired several studies on the boundaries of the achievable QoS for VANETs [12]. Clustering, in this framework, can increase the QoS by providing more connection reliability and increasing connectivity in low coverage scenarios. The Section 15 of this survey provides some examples of data aggregation algorithms in novel scenarios such as IoT, Smart City and VANETs.

Clustering is also envisaged as a tool to provide connectivity in hybrid wireless networks [13]. In a hybrid wireless network, nodes self-organize into a hierarchical (clustered) ad hoc network, and uplink connections are provided by scattered nodes to ensure connectivity in case of network partitions and also to provide internet access. In this scenario, clustering can be used to select the nodes that establish the uplink connections, i.e. the clusterheads, that connect to the base station and provide a gateway for the cluster-members.

Fig. 1 depicts a number of examples of the clustering applications in these emerging scenarios. Smartphones, wearable devices and vehicular technology—cellular networks in general—apply clustering to coordinate the nodes into groups to support efficient cooperative communication strategies. These new scenarios have however yielded new requirements for clustering, that must adopt a multi-objective evaluation strategy for cluster formation, where nodes hardware capabilities, mobility patterns, channel availability, among others, must be taken into account to form clusters. Previously proposed algorithms were mainly based on the consideration of one only parameter such as the node identity, number of neighboring nodes, battery level, speed, etc. In the new scenarios, the network is populated with different technologies (vehicular networks, mobile networks, body area networks, sensor networks, etc.), a variety of hardware capabilities, and the nodes are highly mobile. These technologies are not isolated from each other and it is common to have different devices sharing the same scenario, hence novel clustering techniques must consider a heterogeneous network, where clusters are formed by nodes with different characteristics.

This survey offers a retrospective walkthrough on clustering techniques, with detailed insights on the origins of clustering and the most relevant research efforts during the previous decades. We detail in this manuscript the main merits of previous proposals but also the main drawbacks, and the limitations of these algorithms to be applied in novel areas of wireless networking. The survey also addresses how the new scenarios of wireless networking have fostered a renewed research on clustering, and describes the most recent approaches. At the end, the survey also concludes with the learned lessons that can be drawn from surveying the state of the art in clustering.

2. Organization of the survey

This manuscript is organized as follows: Section 2 defines what is clustering, and provides a thorough explanation of its functionality and main objectives; Section 3 describes the clustering benefits, costs and design considerations; Section 4 describes the most adopted metrics for clustering performance evaluation; Section 5 provides a thorough walkthrough of the origins of clustering; From Sections 6 to 14 the manuscript lists the different clustering categories and describes the most novel algorithms falling in each of these categories. The survey also provides summary tables at the end of each section where readers can find a short description of the algorithms detailed in the text; Section 15 provides a short summary of data aggregation in clustered topologies; Section 16 provides the lessons learned by the authors of this manuscript; Section 17 describes the newest trends and future research directions; finally, Section 18 concludes this survey.

Fig. 2 provides a schematic chart of the main approaches in the state-of-the-art of clustering described along this survey. This figure depicts the main fields, respective sub-fields, and the newest trends of clustering, and includes some references falling in these research areas. In this schematic chart, the tagged squares represent the main clustering categories, from which the respective subfields are extended representing the evolution of clustering techniques along the last two decades. Fields included in circles represent future trends and new topics.

3. What is clustering and how does it work?

Clustering techniques are diverse, but there is a common trend followed by the majority of research efforts that was given by the first publications on clustering. This common approach is based on converting a flat network into a clustered network by assigning roles: clusterhead (leader of the cluster); cluster-member (node belonging to the cluster); and gateway (a cluster-member that belongs to one cluster but connects two clusterheads). The common approach is to select the clusterheads first according to a predefined metric, then the rest of the nodes join one of the surrounding clusterheads and finally the cluster-members that interconnect clusters become gateways. Fig. 3 depicts this procedure.

The clustering formation process requires the transmission of control information. Nodes transmit information such as node identity, list of neighbors, mobility, battery level, etc. This information is required for the clusterhead selection and cluster formation process. The node identity and the list of neighbors is a common control information in all clustering algorithms; other metrics such as mobility or energy, however, are characteristics of some algorithms that target very specific scenarios, such as mobile networks

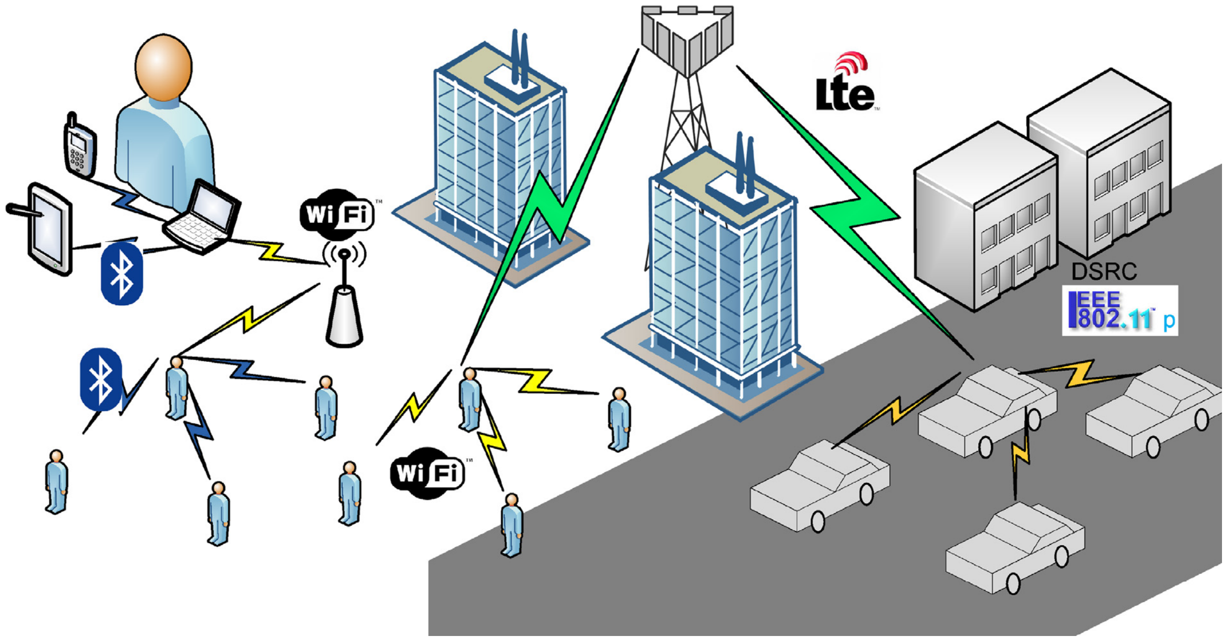

Fig. 1. Clustering for data aggregation in a multi-standard wireless network.

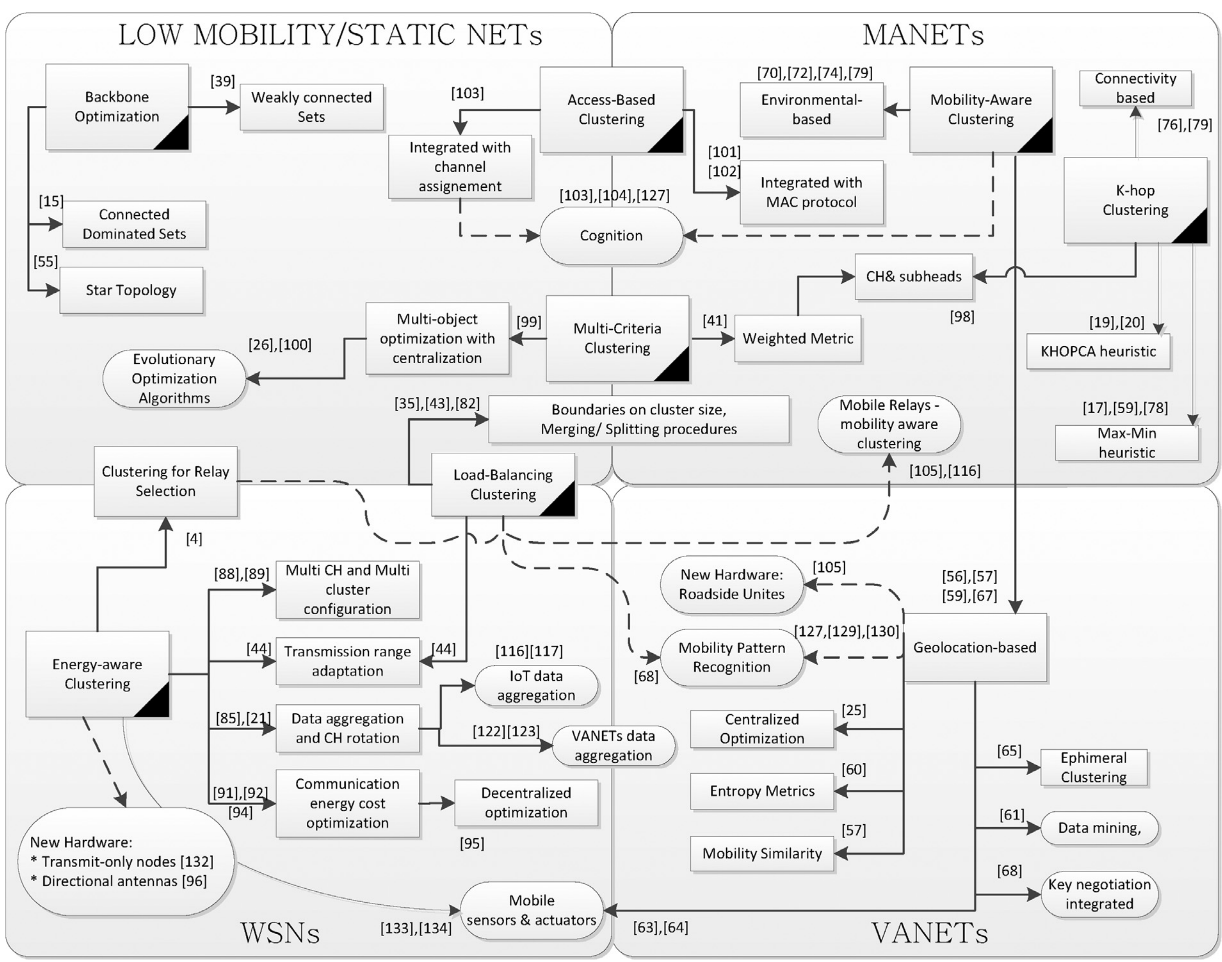

Fig. 2. Schematic chart with the clustering techniques described along this survey. Tagged squares represent the main fields of clustering, from which we extend the respective sub-fields, representing the evolution of the clustering techniques. Fields included in circles represent future trends and new topics. Some references falling in this areas are included in the chart in order to provide the reader with some examples.

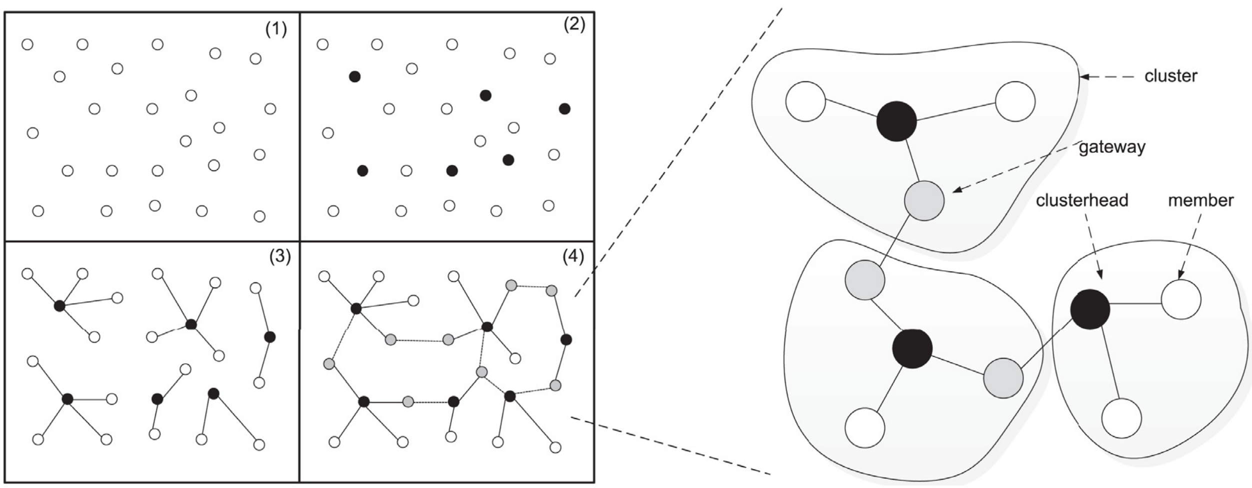

Fig. 3. Steps to set a clustered network from a flat topology, and the roles assigned to the nodes in the clustered network.

and wireless sensor networks respectively. The way this information is used to form clusters is a fundamental part of the clustering algorithms and it will be thoroughly explained throughout this manuscript.

Regardless of the information used to form clusters, a clustering algorithm is defined by a set of rules that compose an heuristic to solve the problem of transforming a flat network into a clustered network. The existing heuristics in the state of the art are diverse, but they follow the same lines of thought, and it is indeed very common to find clustering approaches with the same heuristic, only changing the metrics to select clusterheads. All heuristics are based on the principle of exchanging control information, detect the candidate clusterhead nodes, and then forming clusters. This process however, can be done in different manners to adapt the algorithm to the characteristics of different scenarios. Following subsections explain the different approaches in the state of the art.

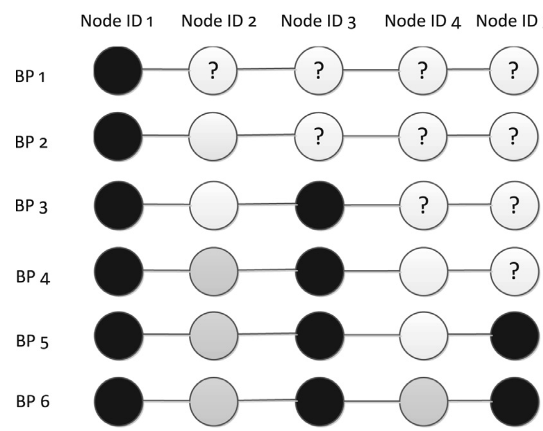

Fig. 4. Toy scenario for an example of cluster formation in a multi-epoch clustering algorithm. The nodes exchange one control message per epoch and progressively form the topology.

3.1. Transmission strategy of control information

Clustering algorithms require a control information exchange. This exchange however may require tight synchronization, loose synchronization, or no synchronization at all. The common approach is a periodic control packet transmission (also called *Hello message* in many research efforts) with loose synchronization, i.e. there is a fixed broadcast period and in every broadcast period the nodes transmit control packets without any specific order for transmission. There is an allowed delay for the packet transmission to reduce packet collision. This approach is the most common due to its simplicity since the nodes only need to set a timer to transmit the control packet every broadcast period. However, this approach yields clustering algorithms for which it is not possible to define a priori number of control transmissions required to get a clustered topology.

Fig. 4 provides an example of a non synchronized clustering algorithm. For this example, we consider a hypothetical clustering algorithm that elects as clusterheads the nodes with the lowest identities (IDs), and transmits only one control packet per broadcast period, but there is not any specific order for control packet transmission. For the sake of simplicity, the nodes are located in a straight line and every node is in contact with its adjacent nodes. In every broadcast period, the nodes transmit their status (role in the topology: clusterhead represented in black, member represented in white and gateway represented in gray) and their ID. After one broadcast period, node 1 realizes it is the node with lowest ID (its only neighbor has an ID of 2), hence it self-elects as clusterhead. In the second broadcast period, node 1 transmits its new status as clusterhead and node 2 becomes its cluster-member. In the third broadcast period node 3 notices that the only node with lower ID is already a member of another cluster, thus it cannot be a clusterhead. Hence node 3 also becomes a clusterhead. It can be appreciated how the nodes with higher IDs have to wait for nodes with lower IDs, leading, in this example, to a delay of 6 broadcast periods for the completion of the clustering topology.

Note that, if the nodes were tightly synchronized, hence transmitting by order of ID, starting by the lowest IDs, then the network would be clusterized in only two broadcast periods. This strategy of transmitting by order was suggested by the first clustering algorithm in [14]. Tight synchronization is also adopted in other algorithms designed for static networks such as the work in [15] that presents an algorithm that achieves a complete hierarchical topology in two broadcast periods. Majority of algorithms however, do not rely on tight synchronization approaches. Loose synchronization clustering is effective in the majority of scenarios, and it converges if the periodicity of control information exchanged is tuned in concordance with the changeability of the scenario [16].

Another approach that doesn't require synchronization was suggested in [17]. The algorithm completes the clustered topology in a predefined number of rounds for control packet transmission. Every node must transmit 1 packet per round, but the only premise

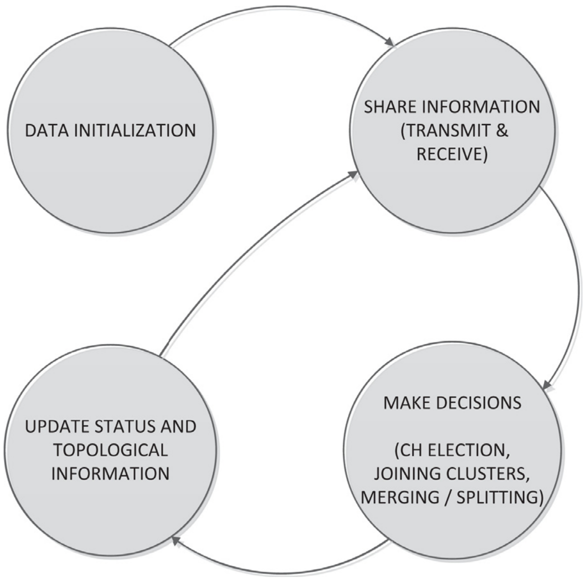

Fig. 5. Behavior of a node when performing clustering related operations with a continuous control information exchange.

to transmit the packet of round i is to have received all packets of round $i-1$ from neighbors. Hence, it is sufficient to tag the packets with the round number and to transmit these packets asynchronously. This algorithm is explained in Section 6.

3.2. Cluster maintenance

Churn can occur due to node mobility, battery limitation, lack of coverage, etc. The clustered topology however must be resilient to eventual node disconnections, hence there is a requirement for a maintenance mechanism. There are three different approaches for cluster maintenance: continuous, event-driven or temporized. The former is the most common approach, suggested for highly changeable networks where the nodes vanish or change location frequently. In such scenario the nodes transmit control packets continuously, following a fixed broadcast period, and clusters are broken and formed locally and on-demand.

Event-driven and temporized maintenance are oriented to static and stable networks. In temporized approaches, nodes do not exchange more control packets after cluster formation. Nodes wait until a time-out expiration, then the nodes re-run the clustering algorithm and form a new clustered topology. The timeout can be substituted by a list of events that trigger the re-clustering process, such as node disconnections or clusterhead overlapping. This event-driven approach can reduce substantially the control overhead in stable networks. This approach however can result in a considerable control overhead exchange in mobile networks, since any single change can trigger a re-clustering process in the whole network.

It is worth mentioning that the continuous clustering maintenance, where nodes send periodically control information, is the most common approach. This has been proven to be more efficient in ad hoc networking, since re-clustering can be locally driven when disconnection events happen, hence making the clustered topology more stable. Moreover, the provision of frequently updated control information allows a faster reaction to disconnection events, this is indeed mandatory in highly mobile networks [18]. In this approach, the clustering formation process and cluster maintenance process are not distinguished, since the nodes are continuously performing the same tasks, resulting in a permanent cluster formation process. This node behavior is summarized in Fig. 5: The

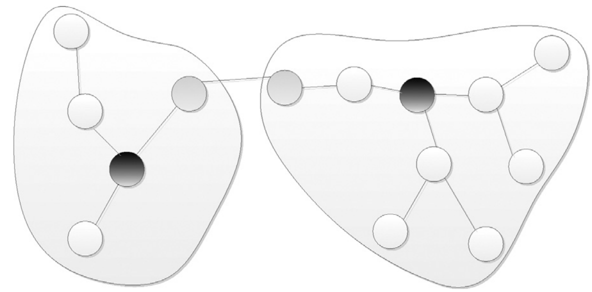

Fig. 6. Multihop clustering; cluster-members are not required to be directly connected to the clusterhead, [17,19].

nodes transmit control information and receive control information from neighbors; then the nodes perform decisions according to the clustering algorithm predefined rules; the nodes update their status and wait for the next broadcast period. The nodes perform the same tasks regardless they are forming the first clustered topology or maintaining it.

3.3. Cluster size

The most adopted heuristic consists of exchanging control information to select candidate clusterheads locally, among the 1-hop neighbors, then nodes join the clusterheads to form 1-hop clusters. As explained before, this can be accomplished in only 2 broadcast periods if the network is tightly synchronized. Some algorithms provide variations for the cluster formation process to achieve multi-hop clustering. The work in [17], Max–Min algorithm, proposed the first heuristic to achieve k-hop clusters in 2k rounds of multi-hop control information exchange. The cluster-members in these approach are not necessarily connected to the clusterheads but can be up to k hops away, Fig. 6. This approach provides more flexibility in the cluster formation process and yields topologies with less number of clusters. The intra-cluster communication cost however, in terms of energy and delay, is increased due to the number of hops between cluster-members and clusterheads.

The heuristic to achieve k-hop clusters, proposed in the Max–Min algorithm [17], is explained in Section 6 of this survey. Subsequent multi-hop clustering algorithms in the state of the art use the same heuristic with different metrics for clusterhead election. Also, more novel works such as KHOPCA, [19] and [20], propose a more recent and novel heuristic to provide k-hop clustering, which only requires single-hop control packet exchanges. This new heuristic is detailed in Section 10.

3.4. Clusterhead role assignment

The clusterhead node bears some extra tasks, such as intra-cluster communication coordination or inter-cluster routing, that requires extra energy consumption. This unbalanced energy consumption is a problem in static networks, since the topology is stable and the clusterhead nodes keep their role for long periods, hence draining their batteries faster. In mobile networks however, the breakage events, produced when the clusterhead loses all members, merges with another cluster or the clusterhead vanishes due to loss of connectivity, induce the selection of new clusterheads. Hence, cluster maintenance implicitly provides a clusterhead rotation mechanism.

For static scenarios such as wireless sensor networks, where the nodes are provided with limited batteries, this is an important issue, that is addressed with intra-cluster clusterhead rotation [21,22]. These algorithms provide a cluster formation process as explained before, but there is a timeout to rotate the clusterhead role among the cluster-members. These algorithms are oriented to

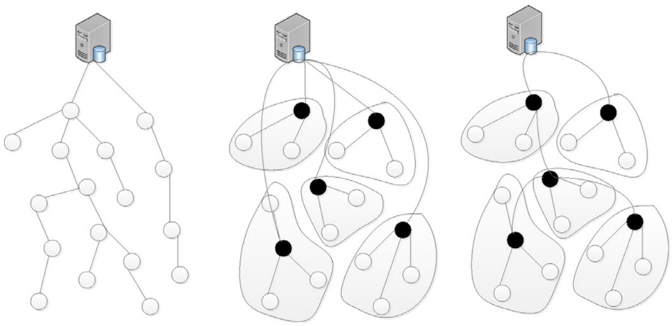

Fig. 7. Scenario depicting a sensor network with a common sink. From left to right: Not-clustered multi-hop connections; clustered single-hop connection; and clustered multi-hop. In cluster-based communications the clusterhead role may rotate among nodes to balance the energy consumption.

wireless sensor networks, where the clusterhead is in charge of aggregating data from cluster-members and relaying this data to the sink. The main idea is to balance the energy consumption by distributing the clusterhead tasks evenly. This clustering approach can be combined with several relay strategies, Fig. 7. It is worth mentioning that clustering for wireless sensor networks have been extensively implemented in real scenarios and yielded outstanding results.

3.5. Overlapping effect

Clustering overlapping is an important feature that has driven substantial research effort in the past. When two clusters overlap it means that the clusterheads are within each other transmission range, whereas in non overlapping clusters, the clusterheads are at least two hops away from each other. There are not thorough studies on the interference effects produced in control and data channels due to clustering overlapping, and it would be a valuable research contribution to provide such a study. It is a fact, however, that the majority of clustering approaches, mainly the most recent ones, avoid clustering overlapping. Even in mobile ad hoc networks, merging procedures are suggested to prevent overlapping due to cluster mobility [23].

In order to obtain overlapped clusters Lowest-ID [14] heuristic proposed that nodes transmit control packets with their ID (identifier) and the lowest IDs in the vicinity are candidate clusterheads. The heuristic is explained in detail in Section 6, but it can be explained with the following logical procedure: i) first select the lowest ID node and draw a circle around this node with the length of the transmission range; nodes inside this circle become a cluster, and the lowest-ID node is the clusterhead; ii) if there are nodes left out of this circle then select the next lowest-ID node and draw a circle around it, if there are nodes inside this circle that where not inside any previous circle, then form a cluster where the second lowest-ID is the clusterhead; iii) continue this process until all nodes are covered by at least one circle. It can be noticed that two nodes selected as lowest-ID in any of the rounds can be under transmission range of each other.

Highest-Degree [24] yields a non overlapped clustered topology because it proposed a different heuristic. The logical process performs in the following manner: (i) first select the node with highest degree (number of neighbors) and draw a circle around this node with the length of the transmission rang; nodes inside this circle become a cluster, and the nodes of this cluster do not participate in future iterations of this heuristic; (ii) if there are nodes left out of this circle then select the next highest-degree node and draw a circle around it; all the nodes lying inside this circle join the clusterhead and form a cluster; (iii) continue this process until all nodes are covered by at least one circle. In this case, since the nodes forming a cluster do not participate in following rounds of the heuristic, it cannot happen that two clusterheads are adjacent nodes.

This difference between overlapping and non overlapping clusters can be appreciated in Fig. 9 (lowest-ID shows an overlapped clustering topology and Highest-Degree a non overlapped topology) and Fig. 11 (CDS and WCDS present overlapped and non overlapped topologies respectively).

3.6. Decentralized versus centralized clustering

The reader may notice that clustering algorithms are decentralized, since nodes transmit information in an ad hoc fashion, and perform an heuristic to achieve the clustered topology. There is not a central entity to retrieve and process the information and to perform the cluster formation task. This is however not true for some of the research proposals, such as [25], [26], that propose more complex optimization protocols for clusterhead selection, hence requiring centralized evaluation of the information and an entity with higher processing capabilities. The former proposes a database and data structures to track the mobility constants of the nodes and predict the nodes locations even with low frequency of control information. A decentralized algorithm would be able to not work so accurately with such a low frequency of control information. The latter provides a multi-object optimization of the clusterhead selection, where the clusterheads are evaluated from multiple perspectives (mobility, degree, energy consumption and distance to neighbors).

The majority of the clustering proposals however are decentralized. It is worth mentioning that authors commonly claim their approaches to be distributed. This is however incorrect in the case of the clustering algorithms described in this manuscript, since the nodes must transmit information and the outcome of the heuristics depend on such information, mainly for clusterhead selection. We should therefore refer to clustering algorithms as decentralized.

4. What are the benefits and the costs of clustering?

In this section, we discuss the most relevant benefits of clustering while highlighting the costs and design considerations of applying clustering techniques. Table 1 summarizes this section (this table is an updated version of the table shown in [27]).

4.1. Benefits of clustering

The benefits obtained from clustering architectures depend on the scenario. Scalability of routing protocols, easier channel allocation and better intra-cluster coordination were the main advantages for MANETs, whereas energy efficiency is considered to be the main benefit for WSNs. In the new scenarios proposed, however, clustering is envisaged to provide stable cooperative relationships, hence the objective is to favor data aggregation. In this regard, clustering offers stable links between cooperating nodes and better coordination to provide contention free intra-cluster communications.

Originally, clustering emerged as a solution for routing scalability problems in ad hoc networking. Ad hoc routing protocols have been proposed by both the research community and standardization groups, such as IETF [28]—and the majority of these proposals result from adaptations of IP-based wired protocols. These protocols however have shown difficulties in adapting to big flat[1] ad hoc networks. When the network is dense, the reactive routing protocols, where routing paths are obtained on demand, incur high

[1] With flat we refer to the absence of hierarchy provided by clustering operations.

Table 1
Benefits and costs of clustering algorithms.

Benefits	
Routing complexity	Sets a smaller backbone formed between clusterheads and gateways to reduce the number of possible routing paths
Interference avoidance	Favors non interfering channel allocation between adjacent clusters. Channel allocation algorithms are easily deployed in clustered networks.
Collision avoidance	Contention-free communication strategies can be deployed for intra-cluster communication. Typically CDMA and TDMA are suggested for communication between culster-members and the clusterhead.
Topological information	The routing information kept by a node in a cluster is limited to the configuration of that cluster. Inter-cluster routing information can be managed by clusterheads and gateways.
Relay selection	Clusterheads in a cluster are assumed to be more stable than the rest of the nodes. They are therefore suitable to be relay nodes.
Coordination	A clusterhead can favor coordination and synchronization for cooperative tasks (i.e. resource management or localization mechanisms)
Costs	
Overhead	Information to form and maintain the topology is periodically exchanged, increasing energy consumption. Periodicity of this information is a trade-off between energy and stability.
Complexity	Apart from the computation complexity, in some algorithms the number of rounds for completeness is not bounded .
Stationary assumption	Information exchanged by nodes to perform the clustering algorithm is supposed to not vary before the completion of the algorithm.
Ripple effect	Strict optimization of the clustering topology can lead to a frequent re-clustering process, thus decreasing stability
Design considerations	
Number of hops	There is a trade-off between stability and communication cost that depends on the number of hops between cluster-members and clusterheads.
Cluster size	There is a trade-off between inter-cluster and intra-cluster communication that depends on the number of clusters and the number of members per cluster.
Inter-cluster connectivity	The number of gateways presents a trade-off involving energy, reliability and routing complexity. The optimum number of gateways depends on the mobility and density of the network.

Table 2
Lowest-ID, transmitted and stored information.

Frame	Node's memory	Node's tx. data
FRAME 1	Neighbors heard	Neighbors heard
FRAME 2	Node status	Node status
	Clusterheads	Connected nodes
	Connected nodes	
	Own clusterhead	

delays. Proactive routing however, where the routing paths are precomputed and stored, incur considerable consumption of resources in terms of overhead and memory [29]. Clustering architectures can reduce routing complexity by reducing the number and size of routes in the network. A smaller backbone is formed by clusterheads and gateways and used to route the packets. Inter-cluster communication may be driven by clusterheads in such a way that members of a cluster just convey the data to the clusterhead. Also, members of a cluster may be configured to only know the nodes belonging to their cluster and default routes pointing to gateway nodes to reach external destinations [30], hence reducing each node routing table. Thus, clustering can effectively reduce routing complexity and make ad hoc networks scalable. In the most novel scenarios, where clustering is applied in cellular networks for data aggregation, routing is not an issue, since the data is transmitted from clusterheads to base stations.

For data aggregation in cellular networks the benefit of clustering is the improved stability of cooperative relationships. Clusters are formed in order to maximize the time availability of clusters, hence maximizing the availability of cooperative relationships. This is imperative to achieve energy efficiency through relaying techniques [4,5].

Another advantage of using a clustering architecture is the interference avoidance obtained from a better channel allocation. A well organized set of clusters allows for an easier deployment of non interfering channels. Typically, in clustered networks, all nodes share a common control channel for cluster formation and maintenance whereas different channels are allocated for data transmission [31]. Thus, communications between different clusterheads and their cluster members do not interfere. Moreover, a contention free communication strategy can be deployed for intra-cluster communications, improving collision avoidance [32].

Load balancing can also be favored by a well organized cluster topology. In [33], a cluster-based network for task-sharing is introduced, where clusterheads can distribute tasks among members according to battery and loads. This idea was also pointed out by authors in [34], where cluster-based networks are suggested to aid service discovery procedures.

4.2. Costs of clustering

Clustering also has its own costs. First, explicit control messages must be exchanged, incurring additional overhead and energy consumption. This information may include topological information (list of neighbors, clusterheads, quality of the links, etc.) and node related information (mobility, battery levels, hardware capabilities, etc.). This signaling is used for neighbor discovery, clusterhead selection, cluster formation and maintenance and also for routing tasks. The periodicity of this message exchange depends on the variability of the network [18,35]. It is worth mentioning that it is possible to piggyback clustering overhead in data packets, as suggested in [36], this approach however assumes that nodes do not have large periods of silence.

The complexity of the clustering algorithm must also be taken into account in terms of computational resource requirements and latency (time required by the algorithm to complete the hierarchical topology). In [37] authors present a study on the complexity of different phases of a clustering algorithm with a graph-theoretical approach that shows polylogarithmic relationship with the number of nodes, links or clusterheads to be selected. This aspect is frequently neglected in clustering algorithms, that normally refer to complexity in order to detail the quantity of information exchange required by the algorithm, instead of the computational effort. In fact, an effective technique to reduce clustering complexity and increase topological stability is to prune not convenient links. This approach is suggested by the authors in [38], which shows that pruning superfluous links between nodes in a static network can increase the clustering coefficient. Clustering coefficient represents the relationship between the number of links in the neighborhood with respect to the maximum possible number of links in the neighborhood, i.e. the maximum cluster coefficient is achieved when, locally, all neighbors are connected directly with each other. Authors in [38] propose an decentralized interactive algorithm, Reckful Roaming, to discard superfluous links. Also the authors in [16] follow the same approach of discarding links. This work shows how short-lived links in clustered networks with fast moving nodes can significantly reduce the stability of the hierarchical network. Although it may seem counter-intuitive, removing links can improve clustering stability and reduce its complexity.

The latency of the algorithm, in terms of time or number of control packet exchanges (broadcast periods), is important since the information exchanged by nodes (energy levels, mobility, list of neighbors among others) has a period of validity. This information, that is continuously varying, influences the cluster formation process. Hence if the cluster formation process takes a considerable time, the information exchange can get stale and lead to a wrong clustering performance. It is worth noting that latency does not only depend on the mathematical operations defined by the algorithm but also the number of broadcast periods required for the algorithm to complete—which sometimes is not bounded. In such case, the information used must be static, such as node's identities or hardware capabilities, and not variable (velocity, battery levels, etc.).

Ripple phenomenon may also occur in a clustering algorithm. This refers to a disturbance effect that a small variation in the network may ripple over the whole clustering topology. Ripple phenomenon may occur when a strict set of rules attempt to optimize the cluster formation in a way that a small variation in one cluster (i.e. a node that loses connectivity) produces a chain reaction affecting the entire cluster topology. As an example, in [15,39] the authors propose two algorithms to set a backbone with optimum size among clusterheads that requires all nodes to be at most one hope away from the backbone's coverage. When one node moves out of this coverage, it triggers a re-computation of the backbone, incurring considerable overheads. Hence, this optimization can be affordable in stable networks where nodes do not move. More variable networks must allow some sort of flexibility in the clustering rules to avoid the ripple effect. In [40] authors address this issue by proposing a one-hop clustering algorithm that grants temporary admission of two-hop members for nodes that suffer eventual disconnections from clusters.

4.3. Design considerations

Some design considerations must also be taken into consideration at the time of applying a clustering techniques, such as the cluster size in terms of number of hops, the number of members per cluster, and the number of cluster interconnections. The effectiveness of clustering relies on how appropriate is the algorithm with respect to the network application.

The size of the cluster, described by the number of hops from the clusterhead (1-hop or multi-hop), and the number of members per cluster, is a design consideration that affects performance of intra-cluster and inter-cluster communications. Big clusters lead to less efficient intra-cluster communication, but it also lowers the number of clusters favoring inter-cluster routing. On the other hand, small clusters can better coordinate contention free mechanisms, hence increasing the efficiency of intra-cluster communication. However small clusters imply higher number of clusters in the network, thus increasing the inter-cluster routing complexity. In this framework, it is important to identify the communication requirements of the wireless technology and the underlying application in order to optimize the size of the clusters. Following this line of thought, authors in [41] propose a load balancing function that measures how well-balanced the topology is, considering the number of clusters and the difference between the optimum and the real number of members per cluster. Authors however do not derive the optimum number of members per cluster since it depends on the network application. Hence, different optimization mechanisms can be used for different scenarios. As an example, authors in [42] propose an optimization protocol to obtain the optimum number of members per cluster in WSNs, that is based on minimizing energy dissipation.

In mobile networks, the cluster size is continuously varying and the optimization is not straight-forward. In this scenario, splitting and merging mechanisms are suggested in [43] to maintain cluster size within some boundaries. Merging consists of joining two adjacent clusters into a single cluster whereas splitting refers to breaking a cluster into two smaller clusters. These techniques are suitable for mobile ad hoc networks with multi-hop configuration, since the clusters cover a wide area and can be easily split into two clusters without producing interference. For 1-hop clustering however, splitting techniques may produce overlap between clusters. Merging is precisely suggested in order to avoid overlaps between mobile clusters. When two clusters move into each other transmission range they merge into a bigger cluster. Thus, merging in 1-hop clustering is not suggested to control the cluster size but to avoid coexistence between clusters. A detailed study on the efficiency of merging procedures in 1-hop clustering can be found in [23]. Another approach is suggested in [35], where authors propose controlling cluster size by applying admission and rejection policies for nodes to join clusters. This technique is effective in sparse networks, but in dense network a big number of rejected nodes may form new clusters that would likely overlap with neighboring clusters. Undoubtedly, the best strategy to control cluster size in 1-hop clustering while avoiding interference is adapting the transmission range. Transmission power adaptation however may not be supported in all wireless interfaces. Transmission range adaptation can be suggested for WSNs [44], since it is provided as a fundamental feature to save energy, but it is not common in MANETs or VANETs.

The number of gateway nodes per cluster is also relevant. Gateways perform a fundamental task in clustered networks by interconnecting different clusters. In this sense, clustering algorithms are usually designed to select gateways based on the nodes' neighborhood. If a node is in touch with at least two different clusterheads, it implicitly becomes a gateway with no further considerations. This however may present a drawback in dense networks, where many nodes are in touch with several clusterheads, leading to the selection of many gateways. Additional extra tasks (topological information delivery and routing tasks) are given to gateways, which not only contribute to drain their batteries but also increase the routing complexity. Hence, the number of gateways can be used for trade-off between connectivity (a big number of gateways improves connectivity and increases reliability) and routing complexity (a small number of gateways reduces the number of routing paths). This issue has driven several research efforts, such as the one in [45], where authors suggest controlling the number of redundant gateways by setting primary gateways and reserved gateways. For this selection the algorithm takes into account velocity, position and lifetime of candidate clusterheads. Notwithstanding, in sparse networks the lack of gateways may lead to isolated clusters. Mobile gateways are proposed to solve this issue in [46], where authors provide a mechanism for the optimization of gateway trajectory. This however requires the capability to control some nodes' movement.

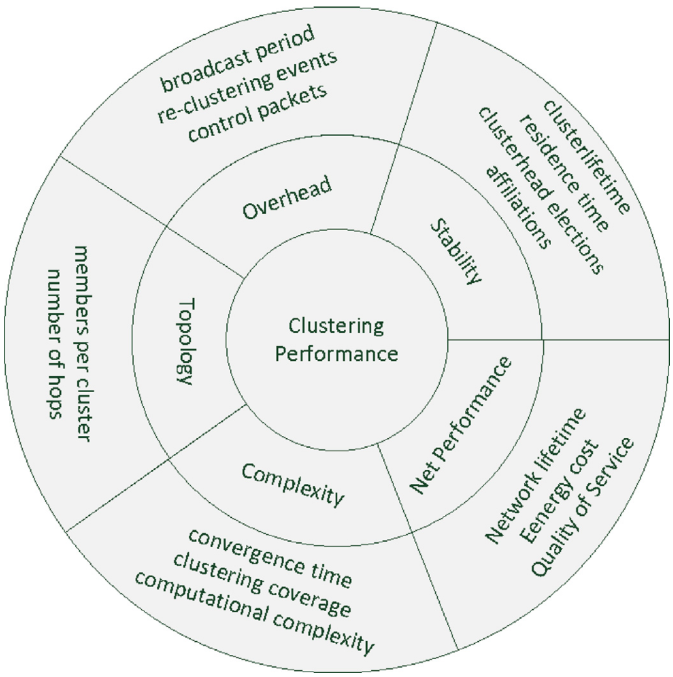

Fig. 8. Commonly adopted cluster performance metrics.

5. How do clustering algorithms compare to each other?

Clustering algorithms are diverse and oriented to different scenarios. Hence, it is hard to determine which algorithm is best, but it is possible to say which algorithm is more suitable for a certain scenario. In this framework, it is important to identify which metrics need to be evaluated to assess the performance of a clustering algorithm. In this section we overview the most common metrics that the readers may find in the performance evaluation of the algorithms cited in this survey. These metrics can target general evaluation of the clustering algorithm—stability, overhead, connectivity—or more specific measurements related with the requirements of the scenario—network lifetime, energy efficiency. Fig. 8 summarizes this section.

Overhead is a main indicator of the complexity of an algorithm, hence it is paramount for a proper comparison to provide an accurate measurement or analytical evaluation on the control message exchange. In algorithms where control information is exchanged continuously, the broadcast period and the control packet structure suffices to evaluate overhead. However, in algorithms where re-clustering is event-driven, it is necessary to track the rate of such events. These metrics can also be substituted by the number of control packet transmissions.

Many algorithms are measured in terms of clusterhead selection and node affiliation events. A clusterhead selection occurs when a node becomes a clusterhead, and a node affiliation is the event of a node joining a cluster. The rate or total number of these events show the stability of the clustered network, hence they are very often used to compare clustering algorithms. These rates are representative of the clustering stability and in some cases the overhead, since node affiliation events or clusterhead selection can trigger dedicated control message transmissions.

Other useful metrics to assess the stability of a clustering topology are the average residence time—the time a node belongs to the same cluster—and the average cluster lifetime—the time a node serves as clusterhead before resigning. It is worth noting that the cluster lifetime limits the residence time, since nodes cannot belong to a cluster after clusterhead resignation. The two metrics, residence time and cluster lifetime, are frequently reported in the performance evaluation of clustering algorithms.

The number of clusters and members per cluster is also a relevant metric to assess how balanced the network is. Ideally clusters should have similar size (number of members), and this size should be close to the optimum number of members (see Section 11). Not only the average but also the variance of these metrics is relevant to assess the homogeneity in the distribution of the nodes among clusters. The network connectivity is also relevant and it can be evaluated with the average of gateways per cluster—nodes connecting different clusters.

Complexity is normally reported as function of the required overhead, but there is as well a time complexity that can be evaluated by simulations, consisting on the time that the algorithm takes to form the clustered topology, also called convergence time. In mobile environments this time may increase due to the frequent cluster breakage events that hinders the completion of the algorithm. It is indeed common in mobile networks that the percentage of nodes into clusters is below 100%, since there are always nodes in the state of looking for new clusterheads to join or even forming new clusters. The average percentage of nodes that are inside clusters is indeed a performance metric that can be called clustering coverage.

More specific metrics may be used to detail the behavior of an algorithm for the special requirements of a specific scenario. In WSNs the network lifetime (usually defined by the time required till the first node runs out of battery) is an essential measurement to evaluate performance. The bit/Joule, comparing the transmitted data and the energy required for the clustering overhead, is also a suitable metric to assess how efficient is the clustered topology. In VANETs the delay for real time traffic is of major importance. Other network oriented metrics like throughput, reliability, routing updates per second, etc. may help to assess the suitability of the topology to provide specific services.

5.1. Common adopted metrics and scenario

Unfortunately, it is not simple to compare existing research efforts in clustering due to the diversity in the simulation scenarios and the different tools to evaluate performance. Authors to not follow the same approach to test their algorithms, and comparisons across different articles are not straight-forward. Moreover, the simulation tools and scenario assumptions vary significantly among papers. The authors of this paper suggest ns2 with the clustering framework in [47] or Matlab with the framework in [48] to undergo detailed simulations on clustering algorithms. We also suggest to evaluate clustering performance in a square-based scenario with random mobility, since it is the most commonly adopted clustering benchmark and it is easier to model analytically.

6. How did clustering research field start?

In this section we detail the first and most influential research proposals in the field of clustering techniques. These algorithms, although obsolete when compared to recent approaches, set the theoretical basis of current proposals. Many of the most recent research efforts in clustering are indeed based on the set of rules described in these first proposals together with more novel metrics for clusterhead selection.

The **Lowest-ID** algorithm [14] was the first clustering algorithm proposed. In this algorithm, nodes are given a unique identifier (ID) that is the only metric used by the algorithm to select clusterheads. The algorithm is periodically executed and each execution is called “epoch” and it is subdivided into two frames in which nodes transmit and receive control information. The transmission strategy

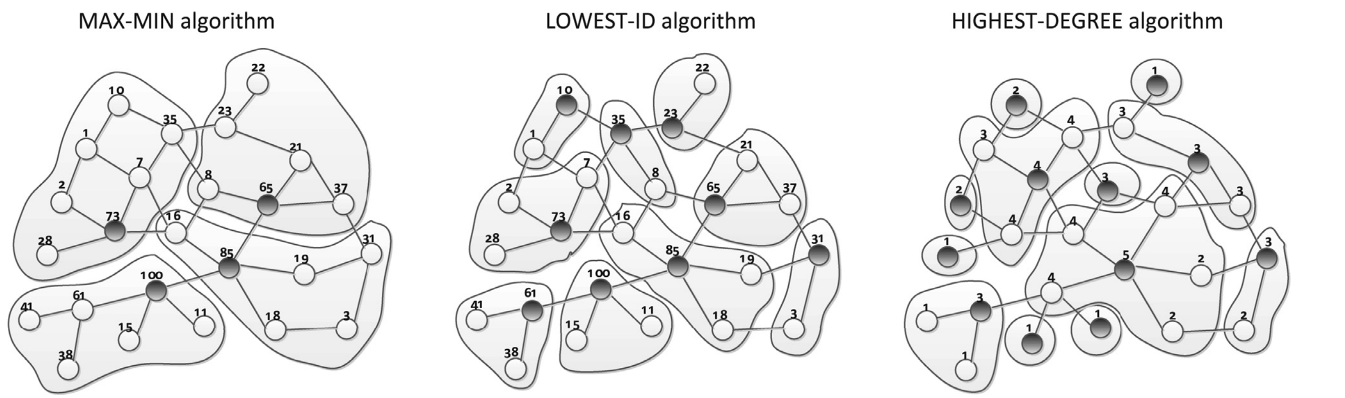

Fig. 9. A clustered topology after applying Lowest-ID, the version in [14], Highest-Degree [24] and Max–Min [17]. For the case of Max–Min and Lowest-ID, the number above the nodes represents the ID of the node, whereas for the case of Highest-Degree the number is the number of neighbors. This topology example is taken from [24], interested readers may find in that paper the details of applying Max–Min in this scenario.

consists of dividing each frame in slots that are given to the nodes to transmit the information in a TDMA-fashion. The nodes must be fully synchronized to transmit in these slots in the correct order, based on their IDs. In the first frame, nodes transmit beacons for neighbor discovery. The beacon includes the list of the nodes heard in previous slots (neighbors that transmitted before). Hence, a node with low ID number transmits and then hears the control packets of adjacent nodes with higher ID. These control packets include the list of nodes heard, hence in this first frame all nodes know whether there is bidirectional connectivity with the neighboring nodes with higher ID. In the second frame the nodes repeat this process, the nodes send the list of nodes with bidirectional connectivity and the node status (clusterhead, cluster-member or gateway). Note that in the second frame the nodes also transmit in order; hence every node, at the time of transmission, is already aware of all bidirectional links with neighboring nodes.

The clusterhead selection process takes place during the second frame. Every node evaluates its connectivity list. If it finds itself as the node with highest ID value in this list, then it self-elects as clusterhead and updates its node status. This node status is transmitted during the second frame. A node also self-elects as clusterhead when it is the node with the highest ID for at least one neighbor; something that the node can check when receiving the connectivity lists from neighbors with lower IDs. The latter can yield a situation where two nodes in contact are clusterheads. This is uncommon in current clustering algorithms that yield topologies where clusterheads are not directly linked. After the second frame however there is a procedure to delete redundant clusterheads, i.e. if all the members of a cluster are covered by any other cluster then the covering cluster overtakes and absorbs all nodes. An example of the application of Lowest-ID is presented in Fig. 9.

The main problem of the above algorithm is the concurrent selection of the same nodes as clusterheads in every epoch. Even if the nodes are mobile, the nodes with higher IDs have higher probability of being selected as clusterheads leading to unbalanced energy consumption. Additionally, the re-clustering process is temporized. It is worth noting that the algorithm uses highest IDs as clusterheads and not lowest, making its name counter-intuitive. This is because the algorithm detailed in this section is an improved version of the paper [49], published the same year, where the same authors use lowest IDs for clusterhead selection. Moreover, a new version is presented a posteriori in [50] where the authors again use lowest IDs. Normally the name of “lowest ID” is well accepted for any of these versions.

Highest-Degree algorithm introduced the concept of variability in the election of clusterheads [24] and asynchronous control information transmission. Unlike lowest-ID, nodes transmit periodically the same message, and without any specific order. During a broadcast period nodes share their list of neighbors, their ID and their current status. After the first broadcast period, all nodes are aware of their neighbors. After two broadcast periods, all nodes know the degree of their neighbors (number of neighbors) and the nodes with higher degree self-elect as clusterheads (in case of tie, the lowest ID gets the clusterhead role). Nodes report their new status in subsequent broadcast periods. Every node in contact with one clusterhead joins the cluster (in case the node is in contact with several clusterheads, it joins the clusterhead with highest degree). Clusterheads and members of clusters are not considered during following broadcast periods of the algorithm. The algorithm converges when all members are either a clusterhead or member of a cluster. Thus, Highest-Degree algorithm does not complete the topology in only two messages per node like Lowest-ID. It requires several iterations, but nodes do not need to transmit in a specific order. It is important to mention that this idea of transmitting the same control message per broadcast period is adopted in most of the existing solutions in the state of the art, including the most novel ones, since it simplifies the clustering formation and maintenance. The disadvantage of the Highest Degree algorithm is that it favors big cluster size, leading to inefficient intra-cluster communication. In addition, topology maintenance is performed with periodic re-clustering in a similar fashion to Lowest-ID. Furthermore, the movement of few nodes may change the degrees of clusterheads leading to a ripple effect. Another problem of this algorithm is that it can produce isolated clusterheads at the border of the topology when the nodes are not well connected, as shown in Fig. 9. This doesn’t happen however if the nodes are well connected, i.e. provided with redundant links such as the example in Fig. 10.

The **Least Cluster Change** algorithm [51] represents a major improvement over previous proposals since cluster maintenance is event-driven. This algorithm is actually a modification of the Lowest-ID algorithm. It follows the same rules for cluster formation, but the re-clustering process is not periodically performed. Instead, there is a list of events that trigger re-clustering, i.e. node disconnection events or clustering overlapping. The topology only changes when two clusters cover each other or a node cannot reach any clusterhead. In such a case re-clustering is performed to form a new topology. This work introduced a new vision/direction for subsequent works on clustering algorithms, since recent approaches do not advocate re-clustering to maintain the hierarchical topology. Instead, maintenance is normally driven locally, minimizing the number of changes in the clustered network.

Another novel idea was given in [17], **Max–Min** since this algorithm was the first providing a technique to form multi-hop

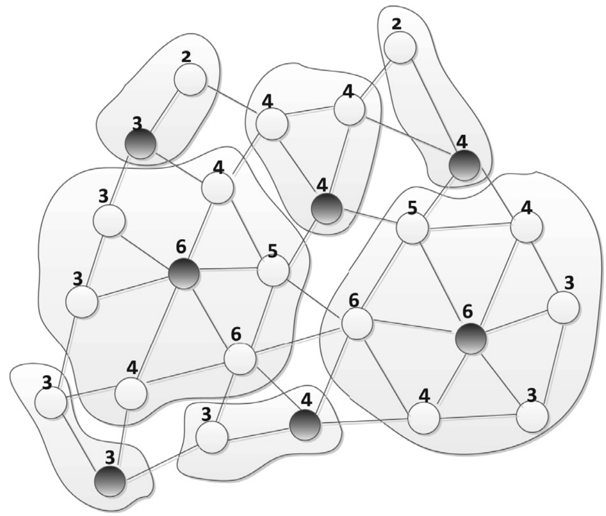

Fig. 10. Highest-Degree algorithm applied in the scenario of Fig. 9, but with more links per node. It can be noticed that there are considerable less number of clusters than in Fig. 9 (7 instead of 12).

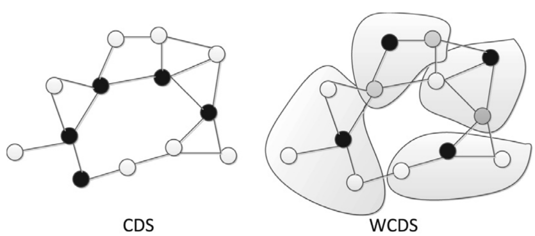

Fig. 11. Connected dominating set and weakly connected dominating set for backbone optimization.

clustering. The previous described algorithms are based on 1-hop clustering where cluster-members are directly connected to clusterheads and at most two hops away from any other member of the same cluster. Max–Min however was the first proposal for multi-hop cluster-based topologies where nodes may be several hops away from their clusterheads. The heuristic proposed in this paper allows for a d-hop configuration where the number of hops , *d*, can be configured without changing the rules of the algorithms, but only the number of messages, 2d messages to complete the clustered topology. Moreover, the algorithm does not require any synchronization. The algorithm is divided in 2 phases of *d* messages, floodmax and floodmin. In floodmax, the nodes select their own ID as a WINNER node (the node candidate to be clusterhead), and transmit a control packet with this WINNER ID. After receiving this first message from neighbors, that can be sent in any order, every node updates its WINNER node with the highest ID among the WINNER IDs received in the control packets. After *d* message exchanges all nodes have their WINNER node set with the highest ID in a range of *d* hops. The nodes keep a list with the WINNER node after each round of floodmax (this list has a minimum of 1 and a maximum of *d* nodes). Floodmin has a similar procedure; the nodes transmit the ID of the WINNER node (the one kept after termination of floodmax), but during floodmin the nodes update the WINNER with the lowest ID among the WINNER nodes received from neighbors. The nodes keep an entry in the list for the current WINNER after each round of the *d* floodmin messages. For clusterhead selection, every node selects itself as clusterhead if its own ID is included in the list of floodmin WINNERs. If not, the node selects as clusterhead to join the lowest ID that appears in both lists, floodmin and floodmax. Finally, if there is no clusterhead selected after the first two rules, then the node selects the highest ID in the floodmax list, that must be the WINNER in the last entry of the floodmax list. This Max–Min approach ensures a balance distribution of the nodes inside clusters. Otherwise, if only floodmax was used for clusterhead selection, the highest ID nodes would have considerably much more cluster-members than lower ID clusterheads.

The readers may notice that some of the concepts about clustering explained in Section 3 are represented in these four algorithms. These algorithms set the basics of clustering and subsequent algorithms adopted some of the previously settled ideas and evolved through more sophisticated clusterhead selection and cluster formation metrics. These metrics respond to the requirement of adapting the algorithms to different scenarios such as sensor networks, vehicular networks or mobile ad hoc networks, that require, in a different balance, the consideration of energy evaluation, mobility, hardware capabilities and so on.

ORIGIN OF CLUSTERING ALGORITHMS
First clustering algorithms. The clustering architecture based on clusterheads, gateways and cluster-members come from these first efforts. The idea of periodic information exchange in order to select clusterheads and form clusters was also proposed by these algorithms and followed by current approaches.

Algorithm	Novelty
Lowest-ID [14]	The first clustering algorithm. It proposes a periodic information exchange between nodes to select clusterheads. Clusterhead selection is performed according the nodes' IDs. Nodes with lower ID get the clusterhead role and the rest of nodes join a neighboring (1-hop) clusterhead.
Highest-Degree [24]	It provides variability in the clusterhead selection and reduction of the number of clusters, since clusterhead selection is done according the number of 1-hop neighbors (node degree). Highest degree nodes are selected as clusterheads, hence the number of clusters is minimized.
Least Cluster Change [51]	The maintenance phase is event driven. It performs like Lowest-ID, but after cluster formation the network is not periodically re-clustered. The network is only re-clustered when two clusterheads get in contact with each other or one node gets out of the clustered topology.
Max–Min [17]	First multi-hop clustering algorithm. It proposes an algorithm where the cluster-members are not necessarily 1-hop away from the clusterhead which provides a more stable topology.

7. How are the clustering algorithms categorized?

Although several clustering classifications have been proposed, in this article we follow the approach provided by [27]. We include however more novel algorithms and new categories. The main categories are: (i) **Backbone optimization**, applied in ad hoc networks where the nodes are mostly static; (ii) **Mobility-aware** clustering, suitable for highly mobile networks and vehicular networks; (iii) **K-hop** clustering for topologies where nodes are several hops away from clusterheads; (iv) **Load-balancing** clustering, suggested for dense networks where the priority is given to balance the effort of the nodes; (v) **Energy-Aware** clustering, that seeks energy preservation in wireless sensor and micro-sensor networks; (vi) **Multi-criteria** clustering, that merges the concepts of previous categories to provide a multi-objective clustering optimization; and (vii) **Access-based** clustering that describes some novel advances where clustering is integrated in medium access control (MAC) protocols or channel assignment protocols. Along the next sections we detail the main innovations in these clustering categories and some incremental research efforts that improved previous ideas.

In order not to burden the reader with redundant information, the following sections focus on the innovative parts of the algorithms, thus skipping the common approaches adopted in the majority of the research efforts, such as the heuristics, that are generally equal or a slight modification of the heuristics explained in Sections 6 and 3. In the following sections we provide insights on the metrics used for clusterhead selection and cluster formation, and the strategies for cluster maintenance. Hence, we focus on the novelty of the algorithms, but we do not detail the algorithms' step by step rules, due to the similarity between the approaches. At the end of each section, there is a summary table with the described algorithms.

8. Clustering algorithms for backbone optimization

Backbone optimization is mainly applicable in static networks, where the topology is stable. In such cases, a complex set of rules, requiring considerable overhead, can be applied to minimize the size of the selected backbone to route the packets. This backbone is formed by a set of dominating nodes (clusterheads) that constitute a unique path along the network where all nodes either belong to the path or are at most one hop away. This path (backbone) is called dominating set and two variations are suggested: connected dominating set (CDS) or weakly connected dominating set (WCDS), as illustrated in Fig. 11.

Wu's algorithm [15] proposes the formation of a CDS with a two-stage algorithm, where a marking process is first performed to select candidate dominating nodes (clusterheads) and then a second process, called prune process, eliminates redundant dominating nodes. The nodes must transmit their IDs and topological information, such as the list of neighbors. Once the nodes are aware of the connectivity metrics of each neighbor, the nodes can start marking themselves as dominating nodes. The self-marking of a node occurs when the node detects that two neighbors are disconnected from each other, in such a case the nodes becomes a clusterhead. Once this phase ends the prune process starts deleting redundant clusterheads. During the pruning process, if a dominating node detects that all its neighbors are already covered by other dominating nodes with higher ID, then it resigns its role. The authors show that this set of rules of marking and unmarking dominating nodes minimizes the backbone size. The complexity of these rules however, in terms of information exchange, have fostered more research work by the authors that suggested in [52] a low-complex prune process for the CDS-based algorithm proposed in [15].

The problem of CDS is the direct connection between clusterheads that leads to an overlapped clustering topology. This is due to the fact that the main intention of CDS is to provide a backbone to support low-complex ad hoc routing. CDS however is not suitable to perform other clustering tasks, such as providing a framework for efficient intra-cluster communication based on TDMA transmission strategies. For this reason, **Chen's algorithm** [39] introduced the concept of WCDS, where the overlap effect is avoided by selecting dominating nodes that are not in touch. The proposed algorithm finds an optimal backbone size that is formed between clusterheads and gateways. The algorithm constructs the topology sequentially treating the network as a collection of pieces. At the beginning every node is a piece. When a node is selected as clusterhead, the node and its neighbors become a bigger piece. When two clusterheads have one neighbor in common, both pieces become a bigger single piece. The algorithm sequentially selects clusterheads that merge bigger pieces together in order to define the whole network as a single piece. The outcome consists of a reduced backbone with non overlapping clusters. These algorithms however, do not take into consideration the number of members per cluster. In fact, the minimization of the number of dominating nodes implicitly favors big cluster size.

Two main drawbacks exist in the previous algorithms: the unsuitability for mobile environments and the unbalanced energy consumption. First, there are no maintenance rules described, thus if the topology is not stable it produces a considerable overhead in re-clustering. On the other hand, if the network is static, the dominating set does not change, leading to an unbalanced energy consumption by the dominating nodes, that will be in charge of the routing tasks. Even if the clustering algorithm is re-executed, it probably leads to the election of the same dominating set. This issue is addressed by Wu et al. in [53], where authors take into account the energy level for the selection of dominating nodes. Thus each time the topology is recomputed the dominating set may change. A similar approach is suggested in **Bhattacharjee's algorithm** [54], that takes into account the degree and the energy levels for the election of the dominating set. The idea is to balance the load of clusterheads by selecting clusterheads with similar number of cluster-members. The latter two approaches however cannot ensure an optimization in the size of the backbone and only offer a nearly optimal size for the sake of a more balanced energy consumption.

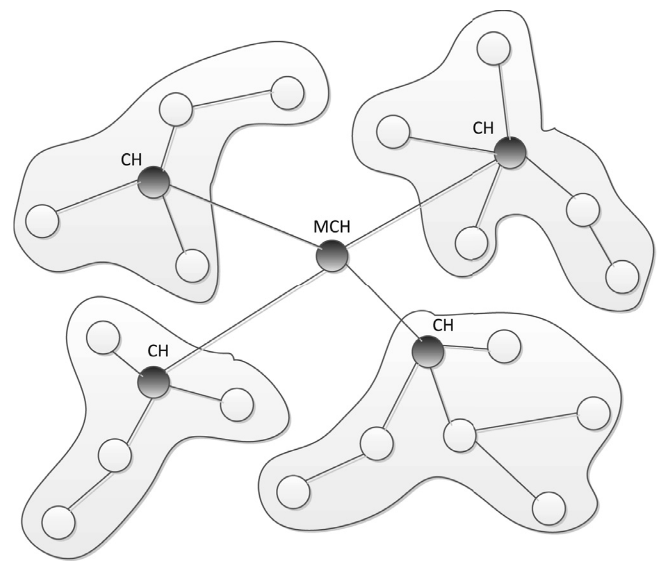

Fig. 12. Start topology as suggested by [55] with a MCH, ordinary clusterheads and clustermembers, which can be connected in multi-hop.

A novel approach for backbone optimization is proposed in [55]. Zahidi et al. propose a centralized algorithm based on Integer Linear Programming (ILP). They propose a mathematical formulation of the cluster formation problem, provided with several constants, such as the number of nodes and clusterheads, and several variables, such as the node connectivity and the roles of the nodes. The algorithm is configured for a fixed number of nodes and clusterheads, and optimizes how the nodes connect to each other and their role. Regarding the nodes' roles, the authors propose a star topology composed of a Master clusterhead (MCH), several ordinary clusterheads and the rest of nodes as members. The MCH connects with all clusterheads and the cluster-members connect to the clusterheads but not to the MCH, see Fig. 12. To achieve such topology, the authors propose a linear utility function and a set of constraints. These constraints enhance intra-cluster connectivity that can be single-hop or multi-hop in order to minimize the communication costs. The fact of considering multi-hop connections from members to clusterheads makes the algorithm flexible to select a prefixed number of clusterheads. Note that if single-hop intra-cluster connection was the only possibility, then setting a fixed number of CHs could lead to some nodes disconnected from the clusterheads for some possible clusterhead set selections; hence the ILP problem formulation would require additional constraints and would be more complex.

BACKBONE OPTIMIZATION
Clustering algorithms for static networks to minimize the backbone to route data. This backbone is formed by clusterheads that can be directly connected, Connected Dominating Set (CDS), or connected through gateway nodes, Weakly Connected Dominating Set (WCDS).

Algorithm	Novelty
Wu's algorithm [15]	It proposes a connected dominating set (CDS), a set of clusterheads directly connected that route the data packets of the network. It provides an algorithm to find the backbone with the minimum size in which all nodes either belong to the backbone or are directly connected to a clusterhead.
Die and Wu [52]	Modification over the Wu's algorithm [15] to reduce the complexity of the second phase of the algorithm, the prune process, in which redundant candidate clusterheads are pruned.
Wu and Dai [53]	It proposes a CDS with clusterhead rotation, since the election of the clusterhead set considers the nodes' energy level. It increases network lifetime but the backbone size is not minimized.
Chen's algorithm [39]	It proposes Weakly Connected Dominating Sets (WCDS) in which the clusterheads are not directly connected but two hops away. There is a gateway node between every pair of clusterheads. This allows clusterheads to form non overlapped clusters.
Bhattacharjee's algorithm [54]	An energy-aware WCDS where the clusterhead nodes are selected according their battery levels and node degree. It produces a more balanced network in terms of energy consumption, but it can only achieve nearly optimal solutions in terms of backbone size.
Zahidi's algorithm [55]	The algorithm uses centralization to achieve a star topology with three levels: master clusterhead, ordinary clusterheads and members. The algorithm uses Integer Linear Programming to achieve an optimal clustered topology. It is the most recent approach in backbone optimization.

9. Mobility-aware clustering algorithms

Mobility-aware clustering techniques aim to address the stability of the clustered topology in mobile environments. Increasing stability refers to maximizing the connectivity time between nodes in a cluster; hence increasing both cluster lifetime and residence time of nodes in clusters.

Mobility information is used as input to predict nodes movement and form stable clusters. These stable clusters are ideally formed by nodes that move with similar patterns (direction and speed) and for which a considerable time of contact is predicted. These algorithms can be categorized regarding the source of information they use. The most common case, mostly presented in VANETs, is the geo-location-based clustering. Nodes are assumed to be equipped with GPS systems and can gather information like speed, direction, position, travel route, among others. Less common but nonetheless important environmental-based algorithms use miscellaneous information provided by the neighborhood, e.g. received signal-strength (RSS) or Doppler effect to predict the time of contact between nodes. These algorithms are designed for networks, where nodes are deprived of GPS or navigation systems, such as wireless sensors or wearable devices. Next subsections describe this categorization of mobility-aware clustering algorithms.

9.1. Geo-location-based clustering

A simplistic approach is given in **WCA (Weighted Clustering Algorithm)** [41], where apart from other parameters mobility is used to select clusterheads. Concretely, a mobility parameter (speed) is calculated by every node. This parameter is computed by gathering the position at different times, by

$$M_v = \frac{1}{T}\sum_{1}^{T}\sqrt{(X_t - X_{t-1})^2 + (Y_t - Y_{t-1})^2} \tag{1}$$

The nodes with lower speed are candidates for clusterheads. This strategy however can only be effective in networks with low mobility. Besides, this multi-criteria algorithm is more influenced by other parameters where load-balancing is usually prioritized, as it can be seen in Section 13.

The idea of group mobility is presented in **MBC (Mobility Based Clustering)** scheme [56]. The authors also use a velocity vector. Every node computes the "Relative Mobility" with respect to each neighbor. The algorithm presents a hierarchical construction of the topology where two metrics are proposed: "Cluster Mobility 1" (CM1) that represents average speed and direction of the Cluster, and "Cluster Mobility 2" (CM2) that represents the uncertainty of the mobility of the nodes belonging to the cluster. The size of the cluster is adapted according to these two metrics, leading to big clusters when nodes move similarly and smaller clusters otherwise. The hierarchical construction indicates that only the stable top level clusterheads are included in the backbone. The less stable clusters are hierarchically linked to the stable ones. The metrics $CM1_I$ and $CM2_I$ for a given cluster I are calculated according to Eq. (2),

$$M_{i,j,t} = \frac{1}{N}\sum_{x=1}^{N}|v_i(t_x) - v_j(t_x)|$$

$$CM1_{I,t} = \frac{1}{M_I}\sum_{k\in C_I} v_k(t)$$

$$CM2_{I,t} = \frac{1}{N_I}\sum_{(m,n)\in C_I} M_{m,n,t} \tag{2}$$

where $v_i(t)$ and $v_j(t)$ are velocity vectors of nodes i and j respectively at time t. Note that M_I is the number of members in cluster I and N_I the number of all possible couples of two nodes in the set of members of I. C_I refers to the set of members of the cluster I.

Similar approach is introduced in **SCP (Stable Clustering Protocol)** scheme [57], as it proposes a metric based on the speed of nodes. Every node obtains its speed vector using GPS and uses the information sent by neighbors to compute the average speed in the neighborhood. Nodes assess their own stability by comparing their speed with the average speed. A "willingness factor" parameter is calculated by comparing the node's velocity with the average velocity and the number of neighbors with similar velocity. If Δ_{vi} is the difference between the average velocity and the velocity of node i and M_i is the number of neighbors that move similar to node i (according to a certain threshold used for comparison), then the stability factor S_i and the willingness factor W_i can be calculated by

$$S_i = \frac{1}{\Delta_{vi} + 1}$$

$$W_i = 2^{S_i log_2(M_i+1)-1} \tag{3}$$

Nodes with higher willingness factor are selected as clusterheads and a threshold over this parameter is used during the maintenance phase to assess if a clusterhead must resign its role. The advantage is that a clusterhead leaving a group can resign before losing contact with the members, hence it speeds up the process of changing clusterhead. The value of this threshold is a tradeoff that must be carefully assessed according to network characteristics. A high value leads to frequent clusterhead re-election, while a low value may not predict the loss of contact with clusterheads.

Also based on group mobility, **Zheng's algorithm** [25] proposes a different metric. The nodes gather their location information in discrete times. If the last position differs by more than a predefined value from the previous one, then the node updates its direction and speed. Nodes compare their speed with neighbors to compute a "Spatial Dependency" parameter that represents the level of mobility similarity between two nodes. Spatial dependency is higher when nodes move similarly. Eq. (4) represents the spatial dependency between nodes i and j at time t.

$$RD(i,j,t) = \cos(\theta_i(t) - \theta_j(t))$$

$$SD(i,j,t) = \frac{\min(S_i(t), S_j(t))}{\max(S_i(t), S_j(t))}$$

$$TSD(i,j,t) = RD(i,j,t) * SD(i,j,t) \tag{4}$$

Note that the vectors θ_i and θ_j and the values S_i and S_j represent the direction and the velocities of nodes i and j respectively. A total spatial dependency (TSD) is defined as the sum of all the spatial dependencies of a node with respect to neighboring nodes (nodes inside the coverage area). The nodes with bigger TSD have bigger number of neighbors and with similar mobility. Hence, they are selected as clusterheads.

A centralized solution is described in **Jensen's algorithm** [58], where nodes report their location information to a server that is considered to be in the origin of coordinates. Each node is considered a moving object O and described by a vector $O(id, x, v, t)$ formed by its identity, position, velocity and the time at which this information was updated. Clusters are also described with a vector of parameters $C(n, cx, cx^2, cv, cv^2, cxv, t)$ composed respectively of the number of nodes, average position, average of the square of positions, average velocity, average of square of velocities, average of vector with velocities and position, and time-stamp of last update. The main reason for these data sets is that it is possible to compute new values and update the data sets even if the information is not received during some time. Besides, the inclusion of a new node in a cluster does not require computation of the whole data set. The main benefits of this algorithm are the good accuracy of mobility information and a low rate of information gathering. The main problem is the centralization requirement, since all nodes must be connected to a server.

KCMBC (K-hop Compound Metric Based Clustering) algorithm [59], deviates from the general idea of using location and mobility information, since clustering is based on predicting link expiration time. Although the estimation of link expiration is still obtained by comparing the speeds of neighboring nodes. The average link expiration time per node is calculated and the nodes with more stable links are selected as clusterheads. The cluster topology is formed following the same approach as Max–Min for k-hop clustering (Section 6). KCMBC also proposes a dynamically adaptive broadcast period to share information with neighbors according to their mobility. It is worth mentioning that the duration of the broadcast period poses a trade-off between overhead, that increases with the decrease of the broadcast period, and stability, that also increases with the decrease of the broadcast period due to the mobility of the nodes. Authors in this paper propose a mechanism to adapt the broadcast period dynamically according to the variability of the network.

The previously described algorithms are based on predicting mobility and computing similarities between neighbors. These metrics however, tend to favor big sets of nodes with similar mobility disregarding the effect of small sets of nodes with different mobility (lower or faster) that lead to high re-affiliation rates (nodes leaving and joining clusters). Thus, previous approaches select clusterheads in order to increase cluster lifetime disregarding the residence time of cluster-members. This issue is addressed

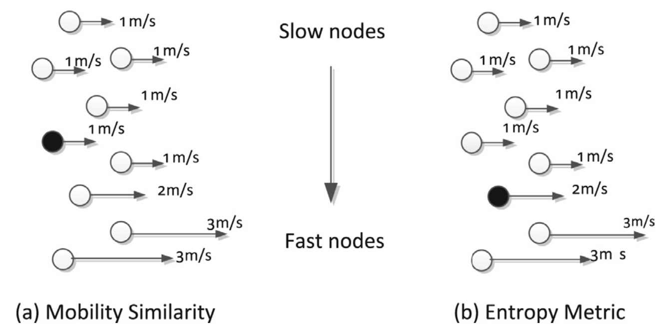

Fig. 13. Clusterhead election based on mobility similarity and entropy metric. (a) part of the cluster is really stable while some minor part of nodes leave and join clusters frequently. (b) clusterhead is similarly stable with respect to all the nodes in the cluster, decreases disconnections during cluster lifetime, but decreases cluster lifetime as well.

in **Wang's algorithm** [60], which proposes an entropy-based metric (Shannon entropy) to assess how ordered is the neighborhood of each node. The mobility values are treated as a distribution of probabilities and nodes with more balanced distribution get higher entropy. Hence, these nodes are selected as clusterheads. Authors define the velocity vector of a node i as v_i and a neighbor j as v_j, which belongs to the set of neighbors of i, N_i (where $|N_i|$ is the number of neighbors of the node i). Then the entropy value of node i can be obtained as

$$P_i(j) = \frac{|v_i - vj|}{\sum_{k \in N_i} |v_i - vk|}$$

$$H_i = \frac{-\sum_{k \in N_i} P_i(k) \log P_i(k)}{log(|N_i|)} \tag{5}$$

Fig. 13 depicts the difference between the entropy-based approach and the previous approaches based on mobility similarity. In this figure, a set of nodes select a clusterhead (the black node) using mobility similarity, case a, and the described entropy-based metric, case b. In this example there are seven nodes with the same moving direction and with different speeds. In the first approach, case a, nodes select a low mobility clusterhead, because most of the nodes have low mobility, leading to a stable cluster for the majority of the nodes in the cluster. However, several nodes that move fast would leave the cluster soon, and eventually other nodes with higher speed will also join the cluster, increasing the number of affiliations. On the other hand, using the entropy metric (case b) the nodes select a node that is equally stable with respect to all nodes in the cluster. The benefit in case b is a considerable reduction in the number of affiliations of nodes to clusters. The penalty is a reduction in the cluster lifetime. In this example the metric in [56] is used for case a, however this issue is present in many clustering algorithms in the literature.

The work of **Hassanabadi** [61] presents a mobility aware clustering algorithm for VANETs that borrows the concepts of data mining, that consists of categorizing data samples in groups of similarity. In this work, the affinity propagation algorithm [62] is used to form mobile clusters. The nodes exchange messages to follow the affinity propagation procedure, making 1-hop groups of nodes with one representative node elected as clusterhead. To evaluate the nodes as data samples, every node is defined by a similarity metric that takes into account the node position and mobility, which is evaluated as a future position estimation. The algorithm is fully decentralized and proposes a synchronized mode that ensures clustering convergence within a predefined time. However this mode does not ensure a predefined number of iterations. It is the broadcast period that has to be manually tuned to fit the predefined time boundary. To achieve the convergence time, the clusterheads are not allowed to relinquish their status until the algorithm finishes.

Mobility-aware clustering can also be applied in sensor networks when the sensors are mobile. Young-jun et al. propose in [63] a mobility-aware algorithm for mobile sensor networks which novelty is the possibility to fine tune the number of clusters homogeneously with a radial approach. The authors assume that the base station, or sink, is located in the center of the scenario, which is bounded by a circle. Then, the scenario is divided by setting an angle resolution. In each area defined by the angle resolution, the clusterheads are selected and periodically switched according to the nodes velocity and direction of movement. The possibility of fine tuning the angle resolutions provides a graceful manner to modify the clusterhead density while maintaining a homogenous distribution of clusters.

Banimelhem's algorithm [64] authors cater for a different vision of mobility-aware clustering for mobile sensor networks in which the nodes' trajectory is optimized to reduce the network overall energy consumption. The novelty of this algorithm is that the clustering process is not performed according to the nodes mobility. It is the mobility of the nodes that is modified to achieve a near-optimal cluster distribution, minimizing the energy consumption. Clusterhead movement is calculated using a genetic algorithm, and after the selection of the clusterhead position the clusterheads move to their selected positions and form clusters. The genetic algorithm uses a fitness function that takes into account the distance that the nodes need to move and the distance between the clusterheads and the sink, hence reducing the overall moving distance and the clusterhead transmission energy. However, the algorithm does not take into account how to move clusterheads according to the position of the nodes and the node coverage, and assumes homogeneous node distribution.

The **CEIF algorithm** [65] proposes grouping vehicles in clusters to gather, more efficiently, information about the traffic conditions. The interesting contribution of this work is the concept of ephemeral cluster formation, where the cluster formation process is pulled from the sink, i.e. the server receiving the traffic information, but is not maintained after receiving the traffic information. Hence, clusters are formed to fit a purpose and then dissolved due to mobility. Every time the sink pulls more information, the clustering process is repeated and a new clustering topology is formed. Authors assume that the traffic information is not frequently gathered by the sink, hence limiting the overhead required. The advantage is its simplicity and the absence of cluster maintenance mechanisms. The disadvantage is that the clusters must be formed reactively, hence increasing the delay of the information gathering process.

There are other mobility-aware algorithms, dependent of navigation systems, that focus exclusively on VANETs, which have specific features such as double direction movement, path boundaries and speed limits, such as **Kulinski's algorithm** [66]. This algorithm proposes a metric to select clusterheads that takes into account link quality, density of nodes and time of contact with neighbors. To reduce affiliations, nodes are categorized as visitors, candidates or cluster members regarding the time of contact in previous epochs. This assumption holds from the fact that in VANETs, nodes that are in contact for a sufficient time can be considered to move in the same path and direction. Nodes moving in the opposite direction appear and disappear and are considered visitors. The main issue with this approach is the computation of the threshold that discerns between visitors and cluster members. The issue of recognizing the different directions of movement for vehicular networks is addressed in **Rawshdeh's algorithm** [67]. The solution is based on previous knowledge of the path, speed limits and speed average. In such a case a threshold can be established to distinguish nodes that move within the same paths. Every node computes its velocity and compares it with each neighbor. If that difference exceeds the threshold, the neighbor is assumed to be moving in opposite direction. However, if the previous knowledge is not given, the authors do not provide a mechanism to dynamically compute such threshold.

Also oriented to VANETs, **Caballero-Gil's algorithm** [68], proposes to integrate into the clustering process a key establishment protocol to provide secure (confidential) intra-cluster data transmissions. Authors propose a clustering algorithm for VANETs which is based on relative mobility and location, similar to previous mobility-aware algorithms. The novel contribution is the integration of the generalization of Diffie-Hellman for multiple users into the cluster formation process. When the cluster is formed, the initial cluster-members send a committed value that forms part of the key, hence preventing the clusterhead from selecting the key itself. Cluster-members joining the cluster after the secret key is computed receive the key through public key encryption. Hence, public keys are only used for key distribution, and shared key encryption is used to provide efficient intra-cluster confidentiality. This algorithm is an example of how clustering can be integrated with other mechanisms that also relay on ad hoc communication, such as key exchange protocols.

An interesting work about the stability of 1-hop clustering for VANETs is presented [69]. In this work, **K. Abboud et al** provide an analytical model to estimate the inter-vehicle distances, residence time of nodes into clusters and cluster overlapping time. The model derived in this work can be used to estimate the achievable limits, in terms of stability and connectivity, of 1-hop clustering for VANETs. Also the work in [23] focuses on the analytical evaluation of clustering overlapping for 1-hop mobile clusters, and assesses the effectiveness of common approaches based on merging procedures (merging overlapped clusters into one).

MOBILITY-AWARE GEO-LOCATION BASED CLUSTERING

Mobility-aware algorithms devised for vehicular networks and highly mobile ad hoc networks. Clustering algorithms are based on geo-location, where position, velocity, map routes, traffic conditions or speed boundaries can be obtained from navigation systems, thus it can be used by clustering algorithms to form stable hierarchical topologies.

Algorithm	Novelty
MBC (Mobility Based Clustering) [56]	The algorithm proposes two metrics to evaluate the mobility level of nodes and the mobility level of clusters. Cluster size is adaptive to its mobility level, yielding bigger clusters in stable networks.
SCP (Stable Clustering Protocol) [57]	Nodes compare their speed to the average speed among their 1-hop neighbors to predict contact loss and select stable nodes as clusterheads. Re-clustering process can be performed before *contact lost* events.
Zheng's algorithm [25]	The clustering process is centralized in a server where nodes send their positioning and mobility information. The algorithm predicts nodes mobility even when the nodes do not update their position and mobility frequently.
KCMBC (K-hop Compound Metric Based Clustering) [59]	Mobility information is used to predict link expiration times. Nodes with more stable links are selected as clusterheads. The algorithm also proposes a dynamic broadcast period which varies according to the link expiration time.
Wang's algorithm [60]	It proposes a mobility metric based on Shannon Entropy to select clusterheads. It reduces affiliation events (nodes hoping from cluster to cluster) with the penalty of reducing cluster lifetime.
Hassanabadi's algorithm [61]	The nodes are seen as a collection of data samples, categorized by their mobility. The algorithm uses the affinity propagation data mining algorithm to form groups of nodes with similar mobility.

(*continued on next page*)

Algorithm	Novelty
Banimelhem's algorithm [64]	The algorithm proposes mobile actuators as clusterheads and optimization of their trajectories for efficient cluster formation in WSNs. A genetic algorithm is used to minimize the distance that the nodes need to move and the distance between the clusterheads and the sink.
Young-jun's algorithm [63]	The scenario is circularly bounded and divided by an angle of resolution. The clusterhead selection is performed independently in each division. By fine-tuning the angle resolution, it is possible to modify the clusterhead density while ensuring a homogeneous distribution.
CEIF algorithm [65]	Proposes ephemeral clustering for VANETs, i.e. the cluster formation process is triggered from a traffic monitoring server to gather traffic information, but the clusters are not maintained after receiving the data.
Kulinski's algorithm [66]	Cluster formation is performed by evaluating link quality, time of contact with neighbors and density of nodes. The algorithm avoids forming clusters between nodes moving in different directions by evaluating the time of contact between nodes.
Rawshdeh's algorithm [67]	It uses knowledge of the path, speed limits and speed average of nodes to set a threshold to discern nodes moving in different directions and form clusters independently in each direction. It is more accurate than other approaches but requires a previous set up.
Caballero-Gil's algorithm [68]	The clustering formation process integrates a key exchange procedure based on Diffie-Hellman. The resulting secret key is used to provide confidential intra-cluster communications.

9.2. Environmental-based clustering

GPS-based algorithms are suitable for VANETs, since vehicles are provided with navigation systems, and for MANETs where nodes are equipped with GPS systems, such as smartphones. However, not all mobile devices are equipped with a GPS; this is the case of wearable devices, wireless sensors, body area network components and legacy handsets. Moreover, the GPS energy consumption disfavors its usage for clustering in smartphones. GPS systems also present limited accuracy in indoor environments. This was the main motivation to develop clustering algorithms based on environmental information, like received signal strength (RSS), signal frequency (Doppler effect), topological information and network layer information.

The first signal strength based clustering algorithm is the well-known **MOBIC** [70]. MOBIC uses the periodic nodes' control messages to compute the relative mobility. The variability of the signal strength in two consecutive Hello messages is used to compute the relative mobility (RM) of a node with a neighbor. Every node can calculate its mobility level, called mobility prediction (MP), as the variance with respect to zero of the set of relative mobility values of that node with respect to its neighbors. The nodes with lower mobility prediction are assumed to move in a similar manner to the neighboring nodes and are selected as clusterheads. However this innovative approach does not have any mechanism to mitigate the effects of fading and multipath when assessing the RM. Another issue of the proposed metric is the variability of the RM value with respect to the distance. As illustrated in Fig. 14, the RM between two nodes with constant speed does not have a constant value. RM is lower when nodes are far and higher when nodes get closer, which is completely counter-intuitive. It is even possible to miss-detect a node as static when it is actually moving. This occurs when two consecutive Hello messages are sent from different positions at the same distance. Eq. (6) defines the RM_i of a node i

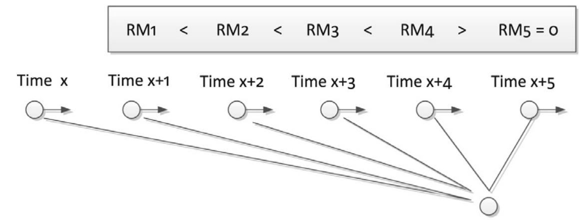

Fig. 14. Variation of the RM values with respect to the distance between nodes in MOBIC.

w.r.t. a neighbor j and the MP_i of i where $\{j_1, \ldots, j_n\}$ is the set of neighbors of node i.

$$RM_{i,j} = 10\log_{10}\left(\frac{RSS_{new}}{RSS_{old}}\right)$$

$$MP_i = var_0(RM_{i,j_1}, \ldots, RM_{i,j_n}) \quad (6)$$

A similar idea is proposed in **Zhong and Ni's algorithm** [71] and [72], where the Doppler effect is used to assess neighbors' mobility. When the transmission frequency is known, the frequency shift received is measured to compute the node's mobility. In the same way as MOBIC, channel characteristics may affect the performance of this algorithm. This algorithm follows the same approach as Mobic for cluster formation and maintenance, that is similar to the Highest-Degree algorithm. Nodes transmit control packets with their IDs, status and list of neighbors. After the exchange of some control packets, the nodes can calculate the relative mobility with respect to neighbors and broadcast such value in following control packets. This value is updated upon reception of a control packet. The nodes with higher stability according to the relative mobility metric self-elect as clusterheads and transmit a CH-claim. 1-hop neighboring nodes can join this clusterhead to form a cluster. In both algorithms, the maintenance phase is event driven, i.e. the control information is continuously transmitted hence updating the mobility metric to form new clusters in case of disconnections.

It is worth mentioning that clustering algorithms based on signal characteristics are often criticized due to inaccuracy induced by fading and multipath effects. The effect of unpredictable channel conditions limits the validity of these mechanisms. New studies on mobility prediction based on channel measurements, such as the one in [73], are fundamental for the application of efficient environmental-based algorithm based on channel related information.

To prevent clustering performance from being affected by channel characteristics some authors propose using topological information to assess mobility. **Gu's algorithm** [74] evaluates the stability of a node, and categorizes a node as static or moving, by evaluating the variability of the list of neighbors. The rate of appearance of any given neighbor and the speed of change of neighbors' identities during a given period of time are used to evaluate the mobility of a node. The drawback is the assumption used in the definition of the scenario: first, the network must be sparse; and second, the nodes must be mostly concentrated in hotspots. Hence, the applicability of the algorithm is limited.

MOBLIST [75] presents a hybrid approach between topological and channel related information. Authors propose a cooperative approach to improve the reliability of mobility metrics based in signal strength measurements. Nodes categorize neighbors as stable or unstable, regarding the similarity of their mobility. These two sets, stable and unstable neighbors, are then compared between nodes in order to assess if the nodes have similar view of the scenario, in such a case the nodes assume that they are moving similarly. The advantage of this technique is that nodes use more information (the whole set of neighboring mobility values) to distinguish which nodes move alike. The main issue of this approach is that there is no mechanism for a dynamic selection of the threshold to discern between stable and non stable neighbors.

A connectivity based approach is presented in **DDCA (Distributed Dynamic Clustering Algorithm)** [76]. The authors propose dynamic adaptive cluster size, where clusters get bigger (in number of hops) when the mobility is low. Although the idea is not new, the innovative contribution is the metric used to assess mobility. Nodes assess the quality of the multi-hop communication with the clusterheads and compute the predicted time of connectivity in a probabilistic fashion. If a node assesses that it can be connected to a clusterhead for at least a time t with a probability α, then it joins the cluster regardless of the number of hops to reach the clusterhead. The fulfillment of this (α, t) requirement is assessed when a node sends a request to join a cluster to a neighbor that already belongs to that cluster. The main drawback of this algorithm is that if the stability of a link in a connected node decreases then this requirement is compromised for other nodes connected through it.

MOBILITY-AWARE ENVIRONMENTAL-BASED CLUSTERING
Clustering algorithms for mobile environments where GPS systems are not provided. Mobility prediction is accomplished by using environmental information gathered during the neighbor discovery phase, such as signal strength or Doppler effect.

Algorithm	Novelty
MOBIC [70]	Mobility is obtained by evaluating the signal strength variations between neighboring nodes. Fading and multipath can affect the accuracy of this approach.
Zhong and Ni's algorithm [71] and [72]	Frequency shift received between two nodes is evaluated to obtain the mobility of the nodes. It is more accurate than MOBIC but also affected by channel conditions.
Gu's algorithm [74]	It uses topological information to evaluate the stability of the nodes. The rate of appearance of any given neighbor, and the speed of change of neighbors' identities is used as an indicator of the mobility of the node.
MOBLIST [75]	Hybrid approach between topological and channel related information. The reliability of mobility metrics based on signal strength measurements is improved by means of a cooperative technique that uses topological information.
DDCA [76]	The nodes evaluate the quality of the multi-hop communication with the clusterheads and predict the link availability time in a probabilistic fashion. This probability is used to form clusters.

10. K-hop clustering algorithms

K-hop clustering is a convenient solution to provide consistent hierarchical topologies in mobile ad-hoc networks since cluster-members do not need to be in direct contact with the clusterhead, hence it limits the reconfiguration events due to node mobility. Authors in [77] show by simulations that, in mobile environments, increasing k implicitly decreases the number of events (clusterhead election and re-affiliations) and increases consistency (time in which the clustered topology remains unchanged in terms of nodes' cluster membership). The increasing consistency however comes with a cost, i.e. increasing the communication hops between cluster members, which increases transmission delay for intra-cluster communication and the inter-cluster interference.

As described in Section 6, the first k-hop clustering algorithm was Max–Min. The proposed heuristic in Max–Min is commonly adopted by other k-hop clustering algorithms that modify some heuristic steps to adapt the algorithm to more specific scenarios. An example of a modification of the Max–Min algorithm is the **KCMBC algorithm** [59], described in Section 9.1, which follows the floodmax-floodmin procedure of Max–Min but with a different

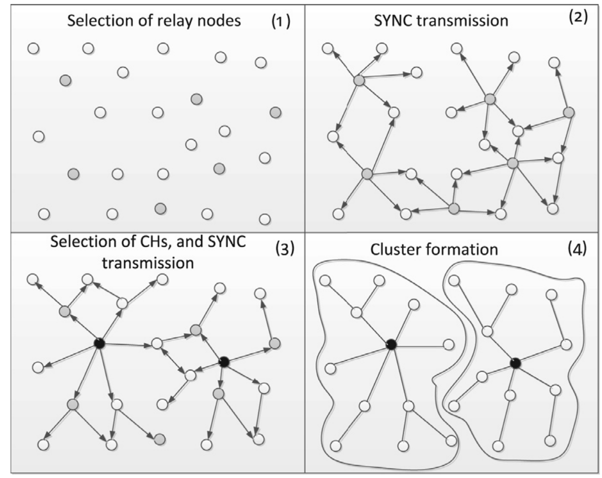

Fig. 15. Example of cluster formation process for the algorithm [80].

clusterhead selection process based on mobility. The **LBKC algorithm** [78] also proposes a modification of the Max–Min algorithm to reduce the overhead. This modification consists in pre-selecting candidate clusterhead nodes to transmit the WINNER packets during the floodmax phase. The rest of the nodes only transmit WINNER packets if they do not receive any WINNER packet from a clusterhead candidate during a period of time. This technique avoids superfluous transmission of WINNER packets. The pre-selection of candidate clusterheads is done according to the nodes ID, prioritizing nodes with higher ID.

The state of the art also counts with multi-hop clustering algorithms that are not based on the Max–Min heuristic, such as **DiLoC algorithm** [79]. In DiLoC the clusters are based on their connectivity. The algorithm uses anchor nodes, established before the algorithm initiation, to trigger the clustering process by transmitting ping messages. Receivers of ping messages join and advertise the cluster. Any unclustered node in contact with a cluster-member joins the cluster of its neighbor. After cluster formation, the anchor nodes become ordinary cluster-members without any special tasks. The cluster maintenance is also based on connectivity and it only triggers new cluster formation in case of loss of connectivity. The algorithm does not provide clusterhead selection mechanisms, and the anchor nodes, which could be seen as temporary clusterheads for cluster formation, must be pre-established. The algorithm also specifies that a WLAN access point can act as anchor node. The advantage of the algorithm is its simplicity, however it lacks any control on the number of hops and it does not establish any node as coordinating point for intra-cluster communications. DiLoC also proposes an upper boundary on the number of cluster members, which can trigger overlapped clusters in dense networks. DDCA [76], described in Section 9, also proposes a multi-hop clustering algorithm without a bounded number of hops between the clusterhead and the cluster members. The limitation is only given by a connectivity metric that estimates the probability that a node is connected to a clusterhead for a given time.

The authors in [80] also present a 2-hop clustering algorithm, **HCA**, that does not follow the Max–Min heuristic. The clustering topology is constructed in three steps, Fig. 15: First the algorithm selects some relay nodes at random, then, in a similar fashion as the clusterhead selection process for 1-hop clustering algorithms, the nodes in contact with the relay nodes become slave nodes; In the second step, the clusterheads are selected at random among the subset of slave nodes that are 1-hop away of at least two relay nodes; In the third step the clusterheads advertise themselves to form a cluster with the surrounding relay and slave nodes. The

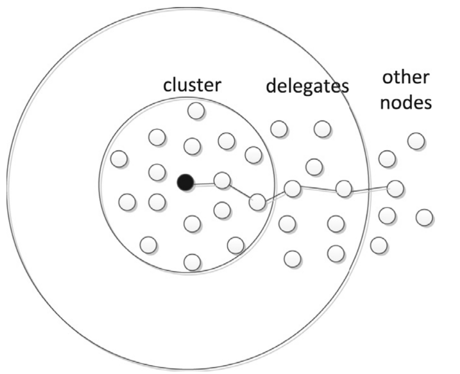

Fig. 16. Example of cluster formation for [81] for k=2 and g=2.

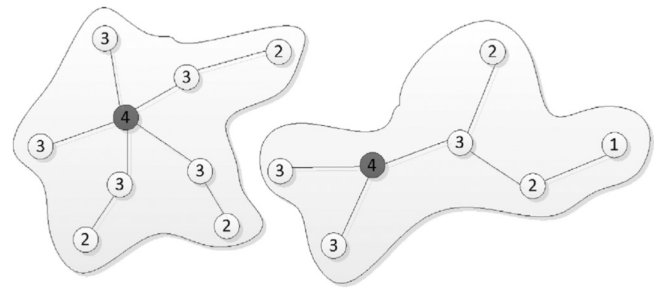

Fig. 17. Example of cluster formation by KHOPCA with k=4.

random selection for the described steps is achieved by dividing each step in time slots. The candidate nodes in each step transmit a SYNC message in one slot selected at random. Candidate nodes that transmit the Sync message become relay nodes or clusterhead nodes, depending on the step, whereas the nodes that receive a SYNC message before their selected slot resign the scheduled transmission and become slaves. The algorithm differs from the Max–Min procedure for cluster formation, mainly because it does not select the clusterheads in its first step, but it also achieves a clustered topology with a bounded number of steps. This heuristic however is only valid for 2-hop clustered topologies and it cannot be extended to other multi-hop configurations.

The work in [81], **Angione's algorithm**, presents a k-hop cluster formation process where the clusters are formed sequentially. First, one node takes the role of initiator and becomes a clusterhead by broadcasting a packet with a TTL (time to live) of g, where $g > k$ and k is the number of hops of the cluster. The nodes receiving the packet with TTL> $(g - k)$ become cluster members, if TTL< $(g - k)$ then the node becomes a delegate and forward a packet to the initiator. The initiator then selects the next clusterhead from the list of delegates by applying some priority rules. Once the next clusterhead is selected, it repeats the process to form a cluster and select another delegate. If eventually a clusterhead does not find delegates, then the process goes backward and previous clusterheads try to find delegates. The clustering process finishes when there are no more delegates. This clustering formation process saves overhead and allows a fine tuned selection of the number of clusters by adjusting the value of g, i.e. the distance between two clusterheads, which is between k and 2k. The drawback is the time required to complete the algorithm, since the cluster formation process is sequential. An example of the cluster formation process for k=2 and g=2 is shown in Fig. 16.

A relevant k-Hop clustering algorithm is proposed in [19] and [20]. **KHOPCA** provides a novel heuristic for the formation of k-hop clusters that, contrary to the Max–Min approach, does not require the transmission of multi-hop control packets. The proposed heuristic only requires one-hop transmission of control packets, hence without the requirement to relay control packets, to form clusters with a maximum of k hops between the clusterhead and the cluster-members. In this algorithm nodes are assigned an initial weight, and the algorithm specifies as input parameters the minimum and maximum weights, which are 1 and k respectively. After the initial weight assignment the nodes' weight is not computed according nodes internal parameters (mobility, energy level, etc.), but modified through the evaluation of 1-hop neighbors' weights according to four simple rules: i) if one neighbor has the highest weight in the neighborhood, then the node selects the same weight but decreased in one and joins the cluster of the neigboring node; ii) if all neighbors have the minimum weigh, then the node adopts the maximum weight and self-elects as clusterhead; iii) If no nodes in the neighborhood have the maximum weight, then the node decreases its weight in one; iv) If two nodes with the maximum weight are in contact, the one with lowest ID decreases the weight in one. The algorithm always converge to a clustered network where clusters have at most k-hops, where k matches with the maximum weight value. Moreover, the nodes know that they can reach the clusterhead through the neighbor with higher weight, and that the number of hops to the clusterhead is equal to the maximum weight value minus their own weight. Fig. 17 shows an example of a clustered network with a maximum weight of k=4.

K-HOP CLUSTERING
Clustering algorithms that form clusters with several hops between clusterheads and members. They provide a more stable topology since nodes can move within the cluster, hence maintaining the cluster memberships longer. However, it increases the intra-cluster communication complexity.

Algorithm	Novelty
Max–Min [17]	First multi-hop clustering algorithm. The algorithm requires two phases of k messages to form k-hop clusters: floodmax and floodmin. It ensures balanced node distribution among clusters.
KCMBC [59]	It follows the Max–Min heuristic but the clusterhead selection process takes into account the mobility of the nodes.
LBKC [78]	It follows the Max–Min heuristic but it makes a pre-selection of clusterhead candidates according to the nodes IDs to limit the transmission of WINNER packets, hence reducing overhead.
DiLoC [79]	Clusterheads, called anchor nodes, are pre-established and they are only responsible of triggering the clustering formation process. Nodes in contact with an anchor node, regardless the number of hops, can join the cluster.
DDCA [76]	There is not a bounded number of hops between members and clusterheads. Nodes join a cluster when there is at least a probability α to be connected during a time t to the clusterhead, regardless the number of relays.
HCA [80]	It forms 2-hop clusters with a three-phase mechanism: first it selects relay nodes; then it selects clusterheads among the nodes that are in contact with at least two relay nodes; 2-hop surrounding nodes join the cluster.
Angione's algorithm [81]	It forms k-hop clusters sequentially. First an initiator forms a k-hop cluster with surrounding nodes, and selects a delegate node out of the cluster to form a new cluster. The process is repeated until the whole network is clustered.
KHOPCA [19], [20]	Proposes a novel heuristic for the formation of k-hop clusters that does not require multi-hop control packet transmissions, i.e. the nodes only exchange control packets with their 1-hop neighbors.

11. Load-balancing clustering algorithms

In dense and low mobility networks the priority is not given to the stability of the clustering topology but to balancing the size of clusters. Cluster size presents a trade-off between inter-cluster and intra-cluster communication performance, see Section 4.3. Moreover, big clusters have a considerable impact on the clusterheads' energy consumption, hence many proposals try to optimize the clusterhead selection in order to achieve a set of clusters with optimum size. In this section we outline the clustering algorithms proposed in literature to achieve balanced hierarchical networks.

WCA algorithm [41] (detailed in Section 13) presents a multimetric clustering scheme that includes a metric to determine how balanced a clustered architecture is regarding the number of members per cluster, Eq. (7);

$$LBF = \frac{n_c}{\sum_i (x_i - \mu)^2} \tag{7}$$

where n_c is the number of clusterheads, x_i is the number of members of the cluster i and μ is the optimum number of clusters. Authors claim that a hierarchical network can be balanced if the clusterhead selection is driven by the maximization of this metric. However, if the network is mobile, the clusterhead election must be re-executed in order to maintain the balance.

Ohta's algorithm [43] proposes a technique to control cluster size that does not require periodic re-clustering. Cluster size is controlled dynamically by including boundaries during cluster maintenance. Two size boundaries, U (upper) and L (lower), are given to trigger merging or splitting procedures. The objective is to keep cluster size between the interval $[L, U]$. In this multi-hop clustering algorithm, every clusterhead submits information about its number of members. If the clusterhead degree C is less than L, then the clusterhead looks for and merges with a neighboring clusterhead with size G such that $L < C + G < U$. Authors show that this approach can effectively maintain the cluster size between these boundaries. This work however does not propose any method to compute the value of these bounds. Moreover, this concept is only suitable in multi-hop clustering, as stated in Section (4).

A different approach is suggested in **Amis' algorithm** [82]. Two parameters, *ED* and *MaxDelta*, are defined representing the optimal number of members per cluster and the maximum allowed deviation from the optimal number of members, respectively. In case the size of a cluster is bigger than $ED + MaxDelta$, the clusterhead resigns its role and breaks the cluster. Re-clustering is then performed by choosing clusterheads that satisfy the "$ED + maxDelta$" boundary. Simulation results probe that this simple directive can sustain the balance of the network. The algorithm however does not describe the procedure in case no nodes can satisfy the boundary, which could bring incompatibilities in dense MANETs.

A more simple approach is introduced by **Gavalas' algorithm** in [35], a subfield "Option" is included in the Hello message. It specifies the current number of cluster members held by a clusterhead. If this number reaches a threshold, then no more nodes may ask for membership of that cluster. Thus no cluster may exceed that threshold. Unfortunately, this approach fosters the creation of overlapped clusters in dense networks and the procedures for the threshold computation are not specified.

In general, these approaches provide low-complex solutions that are effective when the network is not dense. All these efforts lack exhaustive studies on how to calculate the thresholds or boundaries they apply. Moreover, when the size of the cluster is limited by an upper bound and the network size increases, more nodes need to assume the CH role, and this unavoidably leads to overlapping clusters.

In static WSNs however, where the nodes have configurable radio interfaces, cluster size and cluster overlapping can be controlled by adjusting the transmission range. In this framework, **Kawadia's algorithm** [44] performs an optimization of transmission range of nodes that works in networks with both homogeneous and heterogeneous distribution of nodes in the scenario. This optimization seeks the best trade-off between overlapping of clusters (produced when the transmission range increases) and connectivity between clusters (that decreases when the transmission range decreases).

However, it is common in many devices that the transmission range is not tuneable. **Aydin's algorithm** [83] proposes a solution for WSNs, where transmission range cannot be modified. Several clusters can be formed in the same area, but nodes can belong to several clusters simultaneously. In order to balance the load in the clusterheads, the cluster members change periodically their active clusterhead. **Feng's algorithm** [6] uses a different approach based on an optimization algorithm to compute the lowest possible number of clusters needed to cover a specific area. The algorithm explores iteratively the possible solutions to form groups between nodes under two premises, minimization of the number of clusters and an upper bound on the cluster size. The proposed algorithm can achieve more optimal solutions compared to previous approaches but requires centralization. The algorithm is executed in a server that processes the topological information and the nodes are only in charge of sending information to the server and receive the instructions to form the hierarchical topology.

Among the most novel approaches is **Cheng's algorithm** [84], which also proposes centralization, but unlike the previous solutions it takes into account mobility. A social model to describe immigration patterns, Elitism Immigration Genetic Algorithm (EIGA), is used to evaluate the network. The dynamic nature of MANETs is taken into account and modeled as a population, where diversity is given by random immigrants and emigrants representing the fluctuations of nodes in the topology.

LOAD-BALANCING CLUSTERING
Load-balancing techniques are applied in order to obtain clustered topologies where the number of members per cluster is balanced. Several studies propose the application of size boundaries during cluster formation, merging and splitting procedures during cluster maintenance, or the adjustment of the transmission range, which is common in WSNs.

Algorithm	Novelty
Ohta's algorithm [43]	It proposes lower and upper clustersize bounds, in terms of number of cluster-members, with cluster merging and splitting procedures. This approach is suitable for multi-hop clustering algorithms.
Amis' algorithm [82]	It proposes a cluster-member upper bound. If a cluster exceeds the upper bound then the clusterhead resigns and the cluster breaks. New clusters are formed satisfying the mentioned boundary.
Gavalas' algorithm [35]	In this algorithm the clusters have a restrictive upper size bound. When a clusterhead reaches this bound any membership request is denied. Hence nodes left outside clusters must form new clusters.
Kawadia's algorithm [44]	Clustering algorithm for WSNs where the transmission range is adjusted to optimize the trade-off between connectivity and cluster overlapping. The algorithm can work in networks with heterogeneous density of nodes.
Feng's algorithm [6]	The clustering process is centralized in a server. It minimizes the number of clusters while ensuring an upper bound in the number of cluster-members.
Cheng's algorithm [84]	A centralized approach that seeks for a network with optimal cluster size but taking into account mobility. It uses the Elitism Immigration Genetic Algorithm (EIGA), a mathematical model for social behavior.

12. Energy-aware clustering algorithms

Energy-aware algorithms are mostly applied to wireless sensor networks, where mobility or QoS are not a concern. The focus is exclusively given to increase the network lifetime. In this scenario, network lifetime is usually defined as the time elapsed before any node drains its battery. Thus, energy-aware algorithms try to balance the energy expenditure among nodes.

LEACH [85] is one of the most well-known algorithms in this category. In LEACH, the clusterhead collects data from cluster-members and transmits the aggregated data to the sink. To avoid depleting the battery of the clusterhead, this role is periodically rotated among the nodes by assigning random priority values among cluster-members. Thus, data aggregation and clusterhead rotation is the main feature of this algorithm. However, using random clusterhead rotation not always safeguards the best energy efficient topology—such is the case—when the clusterhead role is given to a node with higher energy consumption pattern or lower battery level. This constraint is addressed in **HEED** [21], where the authors propose a vector of parameters to select clusterheads in a probabilistic fashion. Battery level is the primary parameter, but the communication cost with neighbors (in terms of energy consumption) is also taken into account to compute the probability for a node to become the clusterhead. Since battery level is dynamic, the probabilities are periodically recomputed and the clusterhead role is re-assigned. The energy burden of controlling the cluster is fairly shared among nodes.

Other works such as **TL-LEACH** [86] or **Lee's algorithm** [87] also evolve from the concept of LEACH. In TL-LEACH the hierarchical network has three layers: Layer-0, the sensor nodes; layer-1, the clusterhead nodes; and layer-2, the clusterheads that are connected to the sink. The original LEACH algorithm is used to select layer-1 clusterheads and layer-0 ordinary nodes. Then, among the layer-1 clusterheads, the nodes with higher energy are selected to form the layer-2, with the specific task of gathering the sensing information from layer-1 clusterheads and relaying to the sink. TL-LEACH outperforms the original LEACH algorithm in terms of energy preservation. The work in [87] also follows the TL-Leach approach, it provides three hierarchical layers, but the authors introduce a more optimal centralized approach for the selection of the layer-2 heads, called in this work grid heads. The Fuzzy C-Means, a data clustering algorithm, is used to select the grid heads in order to minimize the overall spatial distance between nodes and CHs. After the grid heads are calculated, the conventional LEACH algorithm is used to select the layer-1 clusterheads and form clusters. The algorithm is performed periodically to switch the clusterhead role fairly among nodes.

In dense wireless sensor networks (WSNs), there may be redundancy in the information provided by sensors. **Abram's algorithm** [88] proposes splitting the scenario to several overlapped clusters to cover the same area. These clusters are scheduled in active and non-active periods. At every moment, only one cluster is awake, reporting information, while overlapping clusters turn to a low energy sleep mode. It is worth noting that in WSNs, sensors covering the same area report information that is highly correlated. Thus, the switching off of coexistent clusters do not significantly reduce the information received by the sink. Also following the idea of multi-clusterhead coexistence, Alam et al. [89] suggest to divide the task of collecting and transmitting the aggregated data among two clusterheads per cluster. One clusterhead receives the information from the members and performs the data aggregation task, then it sends the information to the other clusterhead that is in charge of relaying to the sink.

D. Jia's clustering algorithm for WSNs [90] is also based on evaluating redundancy in the sensing area. However, instead of selecting active and non active clusters, the algorithm selects active nodes and hibernating nodes, which can belong to the same clusters. Only active nodes sense data in the region of interest. To distribute fairly the sensing task, the scenario is divided into a hexagonal grid, then an active node per hexagonal cell is selected. After the hexagonal grid is formed, a Voronoi diagram is calculated where each Voronoi cell has an active node as characteristic, i.e. the active nodes are the center of the grid and the redundant (hence hibernating) nodes join the Voronoi cell of the closest active node. Clusterheads are selected among the hibernating nodes to gather the information of the active nodes and relay this information to the sink. The algorithm also defines a scheme to wake up hibernating nodes when clusterheads or active nodes die. The selection of the nodes that wake up follows a metric called Network Perception Contribution, which reflects the redundancy of the node in the sensing area. This technique leads to deplete first the nodes that are more expendable in terms of maintaining the sensing coverage, which increases the WSN lifetime.

Distance between nodes is one of the main parameters affecting the energy consumption, and several proposals focus on this aspect to achieve energy efficient hierarchies. **Manousakis' algorithm** [91] proposes that nearby nodes are grouped in the same cluster in such a way that the overall power transmission is minimized. Inter-cluster communication is also optimized by choosing gateway nodes with the lowest power required. **Ghiasi's algorithm** [92] proposes centralization to perform the optimization of the communication costs. The intermodal distance is the only parameter considered for cluster formation. The sensor network is divided in all possible cluster partitions, where each cluster has the optimal number of nodes. For each cluster, the sum of squared distances with the clusterhead is calculated. The optimal partition is the one minimizing the overall sum of squared distances. Authors approximate the energy consumption by the expression $e = kd^c$, where d is the distance and k and c are specific constants of the wireless system. Thus, distance minimization implicitly means energy consumption minimization. Another centralized approach is suggested in **Bandyopadhay's algorithm** [93]. The optimization in this case is focused on the hierarchy of clusterheads through which the information is routed to the sink. The novelty of this algorithm is the usage of stochastic geometry to perform such an optimization.

Following the approach of considering the link energy consumption, **Parker's algorithm** [94] proposes a centralized approach that evaluates the link energy per bit to noise power spectral density ratio E_b/N_0. The algorithm calculates a weighted graph, i.e. a matrix where each entry represents the weight of the link between two nodes. This weight is calculated according to the E_b/N_0, which also has a minimum boundary that allows discarding links. The algorithm performs a eigendecomposition of the matrix to find the clusterheads, which are the nodes with larger eigencentrality norm and angle. A similar mathematical operation is used to find the assignments between clusterheads and ordinary nodes, which is based on evaluating the angle between the entries of the matrix. Hence, the algorithm solves three problems: determining the number of clusters, clusterhead selection, and cluster formation, with the same mathematical analysis, which is founded on evaluating the links E_b/N_0.

Sohn's algorithm [95] also presents an approach based on reducing the communication energy costs. Authors first propose a maximization problem to define the clustering formation problem, which only considers the link energy consumption between every pair of nodes and between the nodes and the BS. Then, the affinity propagation technique is used to solve the proposed maximization problem in a decentralized fashion. The algorithm is efficient since it does not require centralization, and outperforms previous LEACH-based approaches in terms of energy preservation. The number of clusters is an output of the maximization function, hence it is not required to specify the number of clusters as an input parameter. Finally, the overhead is minimal regardless the size of the network, which makes the algorithm scalable.

Chen et al. propose in [96] a novel clustering algorithm for sensor networks based on directional antennas. Directional antennas are introduced in WSNs to reduce the overlapping effect between sensors in areas with high node density, i.e. areas where there are many sensors gathering redundant data, hence misusing the energy budget. The novelty of the proposed clustering algorithm relies on its consideration of different sectors per sensor. The algorithm considers a first phase, where nodes select a primary sector to sense, and a second phase where nodes select clusterheads and form clusters. The result is a clustered network with the same sensing coverage, but with less overlapping and hence less energy consumption. Another novelty is that the cluster formation process relies on random timers, modeled with a uniform distribution. After timeout expiration the nodes transmit a control packet claiming the ownership of a sector, i.e. following a first come first served fashion. After sector selection, the nodes send control packets informing of their sectors, then the nodes with higher number of neighbors, and surrounded by more primary sectors, are prioritized to be chosen as clusterheads. Of course, the algorithm considers static networks, where sensors are equipped with a switched-beam antenna system.

Although this section focuses on WSNs, the concept of using clustering for data aggregation has been also proposed for MANETs—such as **CONET** [4]. Authors propose to use two different short range wireless technologies—Bluetooth and Wi-Fi. Bluetooth which is considered to be the more energy efficient for short distance communications is used between cluster-members and the clusterheads, while Wi-Fi is used for inter-cluster communications or connection with access points. It can be claimed that it is not demonstrated that the use of both interfaces at the clusterhead would allow improved global energy efficiency, or that the interference caused by the use of both interfaces at the same time (both may operate at the 2.4GHz band) would allow efficient communications. In this framework however, authors in CONET provide experimental results to validate their approach.

ENERGY-AWARE CLUSTERING

Energy-aware clustering constitutes the most prolific field of clustering, since it has produced a considerable number of works with substantial improvement of the state-of-the-art. It is designed for WSNs, where energy saving is fundamental, since sensors are provided with limited batteries.

Algorithm	Novelty
LEACH [85]	It proposes to use clusterheads to gather the sensing information from cluster-members, then aggregates this data and relays to the sink. The clusterhead role is rotated among the cluster-members by assigning random priority values.
HEED [21]	It outperforms the LEACH algorithm in terms of network lifetime, since the clusterhead role is not rotated randomly among cluster-members, but according the nodes battery level.
TL-LEACH [86]	Proposes a variation of LEACH where the hierarchical network has three layers. Layer-0, the sensor nodes, layer-1, the clusterhead nodes, and layer-2, the nodes that aggregate information from the clusterheads and relay to the sink.
S. Lee's algorithm [87]	Proposes a centralized optimization of the node selection of layer-2 in the TL-Leach that improves energy preservation.
Abram's algorithm [88]	This clustering algorithm for WSNs proposes the formation of overlapping clusters that switch on and off alternatively. At any moment, there is only one active cluster per area which is suitable for dense WSNs.
Alam's algorithm [89]	It proposes two clusterheads per cluster. One clusterhead gathers information from cluster-members and performs data aggregation. The other clusterhead relays this data to the sink.
D. Jia's algorithm [90]	It considers three possible states of nodes. Every cluster has a clusterhead in charge of relaying information, an active node that senses data in the region of interest, and the rest of the nodes hibernate. The hibernating nodes wake up selectively to maintain the sensing coverage.
Manousakis' algorithm [91]	The cluster formation is performed in order to minimize the overall transmission power. The only parameter used to form clusters is the distance between nodes.
Ghiasi's algorithm [92]	Centralization algorithm to minimize the communication costs. The sensor network is divided in cluster partitions, where each cluster has the optimal number of nodes. The optimal partition is the one minimizing the overall sum of squared distances.
Bandyopadhay's algorithm [93]	In this algorithm, the communication cost is also minimized. However the authors focus on the multi-hop transmission costs from the clusterheads to the sink. Stochastic geometry is suggested to perform such optimization.
Parker's algorithm [94]	It uses the E_b/N_0 metric to evaluate links and proposes an centralized optimization of the cluster formation process based on a weighted graph.
Sohn's algorithm [95]	It also proposes to the optimize cluster formation process according the links energy consumption. Affinity propagation is proposed to perform the optimization in a decentralized manner.
CONET [4]	Clustering algorithm for MANETs with multi-standard wireless interfaces. Bluetooth is suggested for intra-cluster communication and Wi-Fi for the link between clusterheads and access points.
Chen's algorithm [96]	It proposes to reduce the sensing redundancy by using directional antennas. The selection process of the sensing sectors per node is integrated in the clustering process.

13. Multi-criteria clustering algorithms

Multi-criteria clustering algorithms simultaneously take into account different parameters for cluster formation, such as mobility, location, number of neighbors (degree) or battery levels among others. The first algorithm in this category was the **Weighted Clustering Algorithm (WCA)** [41]. It defines four parameters that are considered in a weighted manner, Eq. (8).

$$W = c_1\Delta + c_2 D + c_3 M + c_4 P \tag{8}$$

In this equation, W represents the node suitability to become clusterhead; lower value gives priority for the clusterhead role assignment. The parameters Δ , D, M and P correspond respectively to: i) the difference between the optimal and the actual node degree (number of neighbors); ii) the average distance with neighbors; iii) the mobility; and iv) the pattern of energy consumption. The vector $C = (c_1, c_2, c_3, c_4)$ is used to weight the parameters (where $0 < c_x < 1$ and $\sum c_x = 1$). The selection of this vector presents the main challenge of this algorithm, since the increase in the importance of one parameter comes on the expense of lowering the importance of other parameters; the selection of the optimal C vector is hence not trivial. The original paper particularly sets a weighting value $c_2 = 0.7$, hence prioritizing load-balancing over the rest of parameters. This limitations have been addressed by novel algorithms, detailed later in this section, through multi-objective optimization techniques.

Despite its limitations, a weighted metric is still a low-complex decentralized solution that has been followed by several other research efforts. One of these approaches is proposed by Wei-dong Yang et al. in [97]. The novelty in this algorithm is the weighted metric, which takes into account 2-hop connectivity and pattern consumption. First the nodes must evaluate their node degree and their neighbor's node degree to calculate the Mean Connectivity Degree (MCD), which consists on the addition of its own degree, d_i, and its neighbors' degree, d_j, divided by its own degree plus one:

$$MCD_i = \frac{\sum_{j \in N_i} d_j + d_i}{(d_i + 1)} \tag{9}$$

Then the metric $D_i = |d_i - MCD_i|$ is defined to represent the node connectivity. Note that a low D_i value means that the node i is surrounded by many neighbors, and that these neighbors are not well connected to other nodes, hence the node i is a suitable candidate to be clusterhead. The algorithm also considers the energy consumption, which is also a function of the node connectivity.

WACA [98] also follows the WCA approach, but proposes multi-hop cluster configuration and a weighted metric aligned with the requirements of hybrid networks [13]. The clusterhead selection process is therefore oriented to find suitable clusterheads to maintain uplink connections. The proposed weighted metric is composed of: i) device power, imperative to maintain the uplink connection; ii) uplink signal strength, to evaluate the performance of the uplink connection; iii) dissemination degree, number of neighbors of the node; iv) local clustering metric, to compare the node connectivity with the network connectivity and avoid clusterhead selection in the network edges; And v) stability coefficient, which evaluates the differences in the node's list of neighbors over the time to estimate the node mobility; One interesting aspect of this approach is the proposition of using two kinds of clusterheads per cluster: One clusterhead, which is in charge of establishing the uplink; and a set of optional subheads that relay information between the clusterhead and the rest of the nodes not connected directly to the clusterhead. The fact that a subhead can directly adopt the clusterhead role in case of disconnection from the cluster improves the re-clustering process. Fig. 18 provides an example of this approach.

When centralization is possible, an improved set of clusterheads can be obtained with optimization algorithms. **Wang's algorithm** [99] uses a Tabu Search optimization to find the clusterheads by looking iteratively for the solutions that have the best outcome in a predefined fitness function. This fitness function is similar to the metric in Eq. (8), where several parameters are simultaneously considered. However the election of the weighting vector is still an issue in this algorithm.

Cheng's algorithm [26] proposes to solve the problem of choosing the weighting vector with a multi-object optimization technique. Clusterheads are selected by using directly the mentioned parameters (mobility, degree, distance and consumption) without any weighting metric. It is based on evolutionary optimization, where a central entity computes random possible clusterhead sets and picks the pareto-optimal solutions. After some predefined iterations, the algorithm searches for better solutions using mutation and crossover techniques.

A multi-objective ant colony optimization algorithm for cluster formation is proposed in [100]. **Chung-Wei's algorithm** considers three objectives, and formulates utility functions to: minimize the number of clusterheads; balance the number of members per cluster (i.e. form clusters with similar number of cluster members); and reduce the nodes energy consumption (i.e. reduce the link distance between nodes belonging to the same cluster). The algorithm uses ACO protocol to find the set of clusterheads and associated cluster-members that minimize these utility functions. The algorithm yields the set of Pareto optimal solutions, and the key innovation is the low complexity search of such set. Such low complexity is achieved with a new solution encoding scheme for the clusterhead set. This clusterhead set is not described by the nodes IDs. Instead, a bit string describes the whole network where a single bit represents each node, and each value is the node's role: 1 CH and 0 CM. Authors claim that the reduced complexity, only $O(2^N)$ with respect to the $O(N!)$ of the state of the art, supports efficient cluster formation of a 300 nodes network in less than a second. However, similarly to previous algorithms, this approach requires centralization.

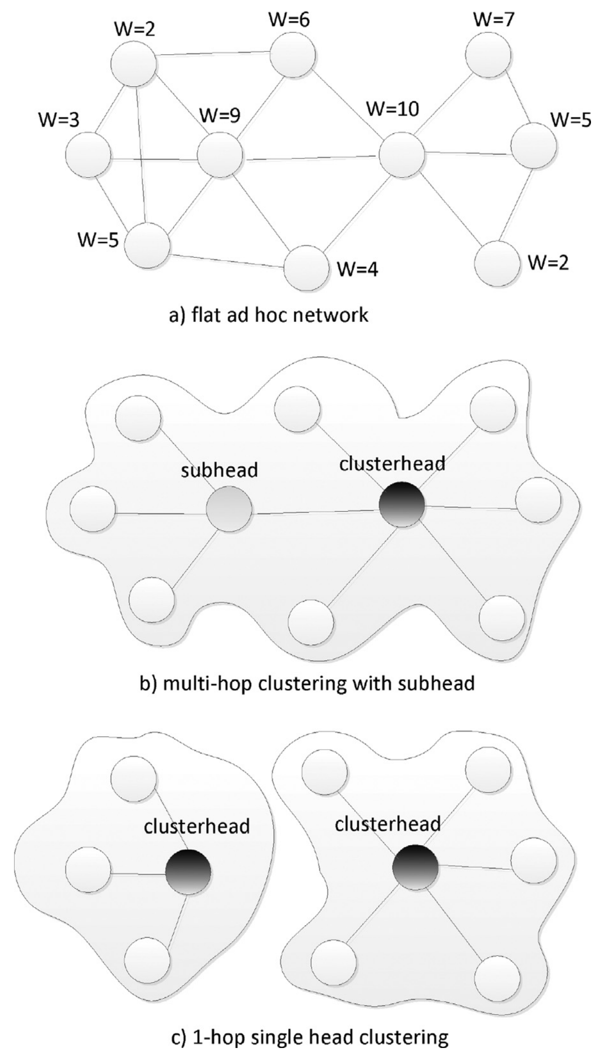

Fig. 18. Example of multi-hop cluster formation with clusterhead and subhead. In the flat network, the nodes are described by their weighted metric. This example is taken from [98].

MULTI-METRIC CLUSTERING
Multi-criteria clustering algorithms consider several parameters simultaneously such as energy, mobility and topological information among others. This approach tries to encompass the characteristics of more specialized algorithms.

Algorithm	Novelty
WCA (Weighted Clustering Algorithm) [41]	First Multi-metric algorithm. It selects clusterheads by taking into account simultaneously four parameters: mobility, number of neighbors (degree), energy consumption and distance with neighbors. The metric is a sum of this four parameters that are weighted according to their relevance.
Wei-dong Yang's algorithm [97]	The weighted metric takes into account 2-hop connectivity and pattern consumption, contrary to previous approaches which only consider 1-hop connectivity.

(continued on next page)

Algorithm	Novelty
WACA [98]	It proposes multi-hop clustering for hybrid networks that introduces the concept of *subhead role*. Subheads are nodes that connect cluster-members to the clusterhead, which overtake the role of clusterhead in case of disconnection.
Wang's algorithm [99]	Similar to WCA but the clusterhead selection and cluster formation process is centralized. This allows a more complex optimization process through a Tabu-Search technique.
Cheng's algorithm [26]	A centralized approach based on evolutionary optimization, where the nodes' parameters are used directly; hence no weighting vector is required. It is currently the most advanced approach.
Chung-Wei's algorithm [100]	Biologically inspired algorithm for multi-metric optimization to find the pareto optimal clusterhead sets. It minimizes the number of clusterheads; balances the number of members per cluster; and reduces the nodes energy consumption.

14. Access-based clustering algorithms

Ad hoc networks are provided with their own MAC protocols and channel assignment techniques, in order to assign different data channels to the clusters and support efficient intra-cluster communication. These techniques are normally deployed independently from the clustering algorithm that forms the topology. However, several authors have suggested to use channel assignment or channel access to form the clusters. Clustering is therefore integrated in the channel assignment or channel access strategy.

Following this line of thought, **Hou's algorithm** [101] proposes that the cluster formation process is tied to the channel access protocol. A CSMA strategy is used in such a way nodes contend for the medium to broadcast control information in a common control channel. During the contention period, the node that gains access to the medium self-elects as clusterhead and the nearby nodes joins the clusterhead. The algorithm provides a detailed channel access-based protocol for which clusterhead election and cluster formation is performed with minimum overhead. Similar approach is used in **Crespo's algorithm** [102], where the authors propose no message exchange for clusterhead selection. When a station needs to transmit, it sends a Request to Send (RTS) message and the receiver becomes a coordinator (clusterhead) for intra-cluster communications (intra-cluster communication is based on Point Coordination Function, PCF). Instead of a Clear to Send (CTS) response, the coordinator sends a beacon to inform all neighboring nodes that PCF communications are possible under its coordination. This temporary cluster will be maintained until a timeout expiration.

The assumption of the availability of a unique control channel is commonly used in the state-of-the-art of clustering. In big networks however different areas and time periods may have different available channels. In this framework, cognitive radio has been suggested for a dynamic selection of the control channel. This is the case of **COGMESH** [103], where the authors propose to use the dynamic channel selection in conjunction with the cluster formation process. This work proposes a biological inspired algorithm, based on Ant Colony Optimization (ACO), to choose and adapt dynamically the common control channel among the nodes. Authors define clusters as the group of nodes under the same channel, thus integrating channel-assignment with cluster formation. The drawback is that it does not provide any control over the size of clusters.

A recent and novel approach is given by Xie et al. in [104]. Authors focus on the clusterhead selection from cluster-members perspective. Commonly, clustering algorithms base the cluster formation process on the election of clusterheads. Once the clusterheads are selected, the cluster members join a clusterhead based on a predefined metric, which is normally sent by the clusterhead, hence equally evaluated by all CMs. In this work however, cluster members select a clusterhead by assessing, individually, the energy consumption required in the link between cluster-member and clusterhead. This strategy has outstanding results in terms of energy efficiency, when clustering is applied for relay selection.

Oriented to vehicular networks, Harikrishnan et al. propose a clustering algorithm for VANETs, where the vehicles form clusters to communicate more efficiently with roadside units (RSUs) [105]. The algorithm's objective is to increase the reliability of V2R (Vehicle to Roadside) connectivity. The nodes in the V2R paradigm connect directly with the roadside units, regardless of the distance or communication cost. In this work however, the authors propose to team up the nodes in clusters, by selecting clusterheads with lower communication costs with the RSUs; hence providing cost-effective communication through a multi-hop link between CMs and RSUs. The metric used to evaluate the communication cost is the path loss. The clusterhead selection process is very simple, consisting in the evaluation of the path loss between a node and the RSU. The nodes join the vehicle in the vicinity with lower path loss, which inherently becomes a clusterhead.

ACCESS-BASED CLUSTERING
Access-based clustering drives the clustering process in conjunction with medium access protocols, hence avoiding clustering-related overhead. The drawback is that mobility, energy or topological information is not considered.

Algorithm	Novelty
Hou's algorithm [101]	The nodes use a CSMA strategy to broadcast control information in a common control channel. During the contention period, the node that gains access to the medium self-elects as clusterhead and the nearby nodes become cluster-members.
Crespo's algorithm [102]	The that want to transmit data send a *Request to Send* (RTS) message, then the receiver becomes a clusterhead. Instead of a *Clear to Send* (CTS) response, the clusterhead sends a beacon to inform all neighboring nodes that PCF communications are possible under its coordination.
COGMESH [103]	Authors define clusters as the group of nodes under the same channel, thus integrating channel-assignment with cluster formation. The selection of the common control channel for clustering operations is done dynamically using the Ant Colony Optimization (ACO) protocol.
Xie's algorithm [104]	Cluster formation is done from the point of view of cluster-members, that join clusterheads according the communication costs. This strategy has outstanding results in terms of energy efficiency when clustering is applied for relay selection.
Harikrishnan' algorithm [105]	The algorithm evaluates the path loss between vehicles and RSUs, and the nodes with lower path loss are selected as clusterheads and perform relaying tasks for the cluster-members.

15. Data aggregation in clustered topologies

Data aggregation has been one of the main drivers for the adoption of clustering techniques, specifically for WSNs. For this reason, in this section, we shed more lights on the data aggregation concept. We provide a brief summary of existing data aggregation algorithms, while concentrating on the application of data aggregation in emerging IoT and smart city scenarios.

Data aggregation was mainly proposed to decrease the amount of redundant data transmitted within WSNs; hence enhancing the energy efficiency of WSNs and thus increasing the network lifetime of WSNs. Data aggregation is generally defined as the process of combining data from multiple sources (sensing nodes) to eliminate redundancy, therefore reducing communication overhead and providing fused information to a sink (external base station) [106]. Since its introduction, many research efforts have addressed data aggregation proposing multiple types of data aggregation algorithms [106,107].

Data aggregation algorithms have been categorized based on the network architecture involved in the data aggregation: (i) Flat networks and (ii) Hierarchical networks. In flat networks, all sensing nodes are considered similar, with the same role in the network and equipped with same capabilities and battery levels, such as SPIN [108] and directed diffusion [109]. In contrast to flat network based, hierarchical data aggregation algorithms involve data fusion at special nodes, based on the properties and locations of different nodes, which significantly reduces the number of messages transmitted to the sink. Under the category of hierarchical network-based algorithms, different categories are defined including cluster-based [21,85], chain-based [110], tree-based [111] and grid-based [112] data aggregation. The first category "cluster-base" uses the concept of clustering to transmit data to a local aggregator (clusterhead), which infuses received data while eliminating redundant data and then transmits to sink. Well-known cluster-based data aggregation algorithms for WSNs are described in LEACH [85], HEED [21] and CLUDDA [113], which is a mix of the two former algorithms. Note that interested readers can find detailed information of these algorithms in Section 12.

Despite the different categories, most of data aggregation algorithms aim at reducing the energy consumption of data transmission by fusing relevant data while eliminating redundant data. In addition to energy efficiency, other performance metrics are usually considered when evaluating data aggregation algorithms, same as in the case of clustering algorithms. Other performance metrics include network lifetime, data accuracy, and latency. All such metrics are important for networks, but usually their priority differs based on the nature of the application; hence there is usually trade-off between different objectives based on such priorities. Recently, with the emergence of Internet of Things (IoT) and smart city applications, WSNs have gained further interest and are more widely used with more sensors than ever, producing more redundant data for higher accuracy. This has surged the interest in data aggregation in smart-city scenarios and applications of IoT.

In the case of Internet of Things (IoT) [114], the process of gathering data may only be practical when data aggregation is possible, due to the battery limitations of these devices. Energy Efficiency in IoT is a cornerstone in the feasibility of a massive deployment of small devices in the smart city scenario [115]. Following this line of thought, several IoT-based data aggregation techniques have been proposed. The authors in [116] consider groups of IoT devices connected to a mobile gateway which is in charge of intermediating between the IoT service and the IoT devices for data exchange, with the main task of aggregating data. In this case the data is aggregated in time and not in space, i.e. there is not a combination of data to eliminate redundant information; the gateway buffers the data during some time and sends all the buffered data to the IoT device. The rationale behind this technique lies in the fact that data buffering allows the wireless interfaces of IoT devices to enter in sleep mode, and wake up when there is sufficient data available. It is worth noticing that this deployment consisting of a gateway connected to several IoT devices is indeed a 1-hop clustering approach, where the gateway acts as clusterhead.

It is worth highlighting the work in [117] that studies the energy expenditure in IoT devices in the uplink. The authors of [117] propose a data aggregation and transmission model based on the selection of data transmitters and aggregators. The algorithm is performed sequentially in several phases. In every phase, some nodes are selected as aggregators in a probabilistic fashion, and the nodes with data to transmit in the uplink send the payloads to the closest aggregator. After transmission of the payloads, the transmitters switch off, and the aggregators perform a new phase of the algorithm. Hence, in every phase there is a new hierarchy of aggregators that receive the payloads from the aggregators of the previous phase. The authors provide a model based on stochastic geometry to select the aggregator density that optimizes energy efficiency.

Novel IoT scenarios, such as crowd-sensing, where mobile devices sense surrounding data cooperatively, aggregate this data, and transmit to a smart-city processing server, have also fostered research on cluster-based security related issues. The work in [118] considers a scenario where devices transmit sensed data over non-secure channels to an aggregator, and provides a mechanism to protect the transmitted data from eavesdropping and data disruption. The proposed mechanism is based on homomorphic encryption with group-based keys, which enables the aggregator/clusterhead to discern malicious/disrupted data transmissions without decrypting. Homomorphic encryption also allows the aggregation of the sensed data without decryption, hence providing end-to-end confidentiality between nodes and the sink. Also the work in [119] provides a framework to evaluate nodes transmitting misleading sensing information based on a trust-based scheme.

Another perspective of the convenience of data aggregation is given in [120], where data aggregation is used to improve the scalability and efficiency of intrusion detection systems (IDS). Conventional IDS are based on inspecting data packets and detecting malicious signature matches; hence it involves an in-depth search in the data transmission flows. This work proposes payload aggregation that consists of selecting a limited amount of bytes (the most significant) in each data flow and aggregating them into the data set to be analyzed. Flow aggregation can also be used for accounting of consumed bandwidth or time, as described in [121].

Intelligent transportation systems also rely on data aggregation to support crowd-sensing for traffic monitoring. On-board sensors are assumed to be provided in future vehicles as well as in RSUs, to track road conditions and traffic dynamics, with the main objective of detecting anomalies in real time, and provide rapid countermeasures. As an example, in [122] the authors provide a system design for such scenario where data aggregation is used to increase the efficiency of large scale data processing and storage. In this work, data aggregation is achieved by making data summaries at the intermediate nodes, i.e. providing minimum, maximum and average values for the parameters of interest (velocity, traffic density, pollution, etc.). Also the work in [123] proposes cluster-based data aggregation for the transmission of traffic information among vehicles. The approach consists on combining the data packets before relaying the packets to eliminate redundant information. Mobility-aware clustering is used in this scenario to form a hierarchy that limits the aggregation task, and the broadcasting transmissions, to certain nodes. Data aggregation has also been proposed for smart city infrastructure [124], namely for video surveillance systems integrated into public transportation. In this case, data aggregation is performed by intermediate gateways, which receive the video streams and perform data pre-processing before transmitting the data to backend servers.

From this brief summary of data aggregation, the importance of data aggregation algorithms is clear in enhancing the energy efficiency, accuracy and scalability of WSNs in the era of IoT, Smart City and Intelligent Transportation Systems. The importance and relevance of clustering techniques to data aggregation algorithms can be easily deduced.

16. Lessons learned

Clustering has been an active field of research over the past two decades, yielding a plethora of algorithms in the diverse sub-fields of wireless ad hoc networking. Some of these algorithms have been successfully applied in real scenarios—such as the case of WSNs, where algorithms like LEACH or HEED have shown outstanding results. Energy efficient clustering for WSNs has seen substantial effort from research community. The endless applications of wireless sensor networking and the requirement for efficient communications, due to hardware limitations, have fostered research on such area.

Clustering algorithms for MANETs have also experienced a surge on proposals addressing different scenarios: static and mobile networks, civil and military applications; and with different purposes: extending network coverage, providing infrastructure-less connectivity on-demand, etc. This research line however has not yielded, yet, the expected results. There is not currently any successful implementation of clustered networks for MANETs, as they were envisioned decades ago. However, the new emerging scenarios, populated with a broad diversity of highly capable wireless devices, have brought about a surge on new proposals for clustered networks that do not focus on clustering as a manner to achieve infrastructureless connectivity, but to provide energy efficiency and scalability, among other features, in cellular networks.

As an evolution of MANETs toward vehicular technology, clustering for VANETs have gained popularity due to its applicability to car safety communications. Vehicles are, in a near future, intended to be equipped with wireless interfaces that will transmit short range control information for traffic safety. DSRC (Dedicated short range connectivity), 802.11p standard, has been indeed designed to support such functionality. In this scenario clustering plays an important role, and the provision of navigation systems and the lack of energy concerns provides an interesting scenario for a successful implementation of clustering solutions.

17. Future work and new trends

Regarding MANETs, the newest trends in clustering go beyond previous literature by including cognition, [125]. In an ad hoc environment, where numerous mobile handsets coexist, cooperative reasoning and learning can be applied. Such is the case for the concept of "wisdom of crowds" [126], which has brought a new vision where heterogeneity and disaggregation of nodes represent an advantage more than a challenge [127]. Its application toward context-aware ad hoc networks requires yet more contributions from research community. Biologically inspired algorithms have also been used as a learning technique for wireless ad hoc networks. The concept of "swarm intelligence" has been applied to clustering in [103] and [100], namely the Ant Colony Optimization algorithm.

Many other optimization algorithms however, based on models for biological behavior, could be applied in this scenario. Such is the case of "morphogenetic robotics" [128], based on the fundamentals of evolution. Authors show in this work that cooperation between independent robots toward the same predefined goal may simply emerge by applying the same mechanisms as in biological morphogenesis. Authors state that the same idea holds for the organization of self-organized networks, but no framework or study is given yet to apply this concept on mobile ad hoc networks.

VANETs is currently the most active research field within the scope of this survey. In this scenario, clustering algorithms benefit from the availability of GPS-based navigation systems and predictable mobility patterns. In such networks, the main challenges are the high speed mobility and the delay sensitive communications. In this sense, additional effort is still required in effective QoS mechanism. Several works have started to point to mobility pattern recognition as a fundamental feature for future vehicular networks [67,75]. Pattern recognition algorithms are however complex, requiring centralization and previous knowledge of the scenario [129] [130]. Research is required in this field in order to integrate low-complex pattern recognition algorithms into clustering, such as the work in [127]. It is worth mentioning that a novel approach in VANETs is proposed in [131], where authors suggest roadside units (RSU) to collect packets and forward them to other RSUs using vehicles as transport units. Thus, RSUs collect and deliver information from and to moving vehicles, and can use some vehicles to transport information to long distances. This way of outsourcing the routing decisions is promising, yet there are few works merging the concepts of vehicular to RSU (V2R) communications and hierarchical networks, such as the work in [105].

Clustering in WSNs has been thoroughly researched in the literature. Energy efficient clustering was mainly triggered by the battery constraints of sensors, and focuses on the efficiency of data aggregation and cluster-based routing mechanisms. However, new advances on the design of WSNs has brought a room for more research efforts. Such is the case of transmit-only-nodes, that consist of sensors with the capability to sense and transmit but not to receive information from other sensors. The approach is to deploy these nodes, that are cheaper, together with other sensors with the capability to receive and transmit. In this scenario, the cluster-based approach has to take into consideration that transmit-only-nodes cannot be clusterheads, just cluster members relaying the information [132]. Moreover these nodes cannot receive signaling from clusterheads, and they are therefore not adaptive for changeable and dynamic algorithms. Actuators are as well a new hardware entity that must be considered in clustering algorithms. Actuators are more powerful nodes with the task of collecting sensing data and, in some cases, with the physical capability to act in the medium according to the data gathered by the sensors. Some efforts have been published suggesting that actuators should take the role of clusterheads [133,134]. It is worth mentioning the work in [134], where authors propose mobile actuators and provide an optimization technique for the actuator's trajectory to collect the data aggregated by sensors. In such scenario, the clusterheads are not selected in a conventional way, with an heuristic that selects clusterheads to cover all nodes, but in a manner that the clusterheads' trajectories can cover all nodes in predefined time intervals.

18. Conclusion

This paper presents a survey on the state-of-the-art in clustering for the different wireless technologies. The specific features and special requirements for mobile ad hoc networks, vehicular networks and sensor networks are identified. The benefits and design considerations of clustering algorithms are exposed. In order to present a clear picture of the state-of-the-art, we detail the main principles of clustering that were set by the first works on this field, and the evolution brought about by the most recent and novel ideas. Clustering algorithms were presented in six main sub-fields: backbone optimization; mobility-aware; energy-efficient; load-balancing; multi-criteria optimization and access-based clustering. The most common performance metrics are also detailed in order to give the reader a hint on how to evaluate and compare different clustering algorithms.

Although clustering has been a prolific field of research during the last decade, there is still room for further contributions within MANETs, WSNs and VANETs. It is probably the latter that is currently receiving most attention due to the emergence of vehicular technology. In this field, new solutions based on hardware implementation, like the inclusion of roadside units, have brought a new vision that requires a re-design of clustering algorithms for

VANETs. Similarly, clustering for WSNs must now consider the network heterogeneity provided by new hardware entities like actuators or transmit-only nodes. In the case of MANETs, mobile handsets are also experiencing technological improvements in terms of computational capacity and the availability of multi-standard short range wireless interfaces. These new capabilities in conjunction with the new advances in context-aware algorithms may foster research on cognitive versions of clustering algorithms, that are more adaptive and efficient.

Acknowledgments

The authors would like to thank the E-COOP project (UID/EEA/50008/2013-FCT/MEC-PIDDAC), and the European Community's FP7/2007-2013 under grant agreement no. 285969 [CODELANCE] that supported this research. Author Ayman Radwan would like to acknowledge the support of the grant of the Fundacao para a Ciencia e a Tecnologia (FCT, Portugal), with the reference number: SFRH/BPD/104585/2014.

References

[1] C.E. Perkins, Ad Hoc Networking, 2001.
[2] DARPA, Survivable Radio Network Project (SURAN), 1980.
[3] DARPA, DARPA Packet Radio Network Project (PRNet), 1973.
[4] J.-W. Yoo, K.-H. Park, A cooperative clustering protocol for energy saving of mobile devices with Wlan and Bluetooth interfaces, IEEE Trans. Mobile Comput. 10 (4) (2011) 491–504, doi:10.1109/TMC.2010.161.
[5] E.U. 7th Framework Project, C2POWER, 2013.
[6] F. Xia, X. Zhao, H. Liu, J. Li, X. Kong, An energy-efficient and load-balanced dynamic clustering protocol for ad-hoc sensor networks, in: 2012 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2012, pp. 215–220, doi:10.1109/CYBER.2012.6392556.
[7] F. Saghezchi, A. Nascimento, M. Albano, A. Radwan, J. Rodriguez, A novel relay selection game in cooperative wireless networks based on combinatorial optimization, in: 2011 IEEE 73rd Vehicular Technology Conference (VTC Spring), 2011, pp. 1–6, doi:10.1109/VETECS.2011.5956688.
[8] H. Cheng, J. Cao, H.-H. Chen, H. Zhang, Grls: Group-based location service in mobile ad hoc networks, IEEE Trans. Vehic. Technol. 57 (6) (2008) 3693–3707, doi:10.1109/TVT.2008.919614.
[9] D. Niyato, P. Wang, W. Saad, A. Hjorungnes, Controlled coalitional games for cooperative mobile social networks, IEEE Trans. Vehic. Technol. 60 (4) (May 2011) 1812–1824, doi:10.1109/TVT.2011.2116816.
[10] G. Remy, S.M. Senouci, F. Jan, Y. Gourhant, Lte4v2x #x2014; collection, dissemination and multi-hop forwarding, in: 2012 IEEE International Conference on Communications (ICC), 2012, pp. 120–125, doi:10.1109/ICC.2012.6364412.
[11] K. Sampigethaya, M. Li, L. Huang, R. Poovendran, Amoeba: Robust location privacy scheme for vanet, IEEE J. Selected Areas Commun. 25 (8) (2007) 1569–1589, doi:10.1109/JSAC.2007.071007.
[12] M. Boban, G. Misek, O. Tonguz, What is the best achievable qos for unicast routing in vanets? in: 2008 IEEE GLOBECOM Workshops, 2008, pp. 1–10, doi:10.1109/GLOCOMW.2008.ECP.69.
[13] A. Andronache, M.R. Brust, S. Rothkugel, Multimedia content distribution in hybrid wireless networks using weighted clustering, in: Proceedings of the 2Nd ACM International Workshop on Wireless Multimedia Networking and Performance Modeling, in: WMuNeP '06, ACM, New York, NY, USA, 2006, pp. 1–10, doi:10.1145/1163698.1163700.
[14] D. Baker, A. Ephremides, The architectural organization of a mobile radio network via a distributed algorithm, IEEE Trans. Commun. 29 (11) (1981) 1694–1701, doi:10.1109/TCOM.1981.1094909.
[15] J. Wu, et al., On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks, 1999,
[16] H. Marques, J. Rodriguez, S. Seiamak, R. Tafazolli, V. Sucasas, A. Radwan, Can we apply clustering in fast moving objects? in: The Eighth International Wireless Internet Conference (WiCON 2014)—Symposium on Wireless and Vehicular Communication, Lisbon, Portugal, 13-14th Nov. 2014, 2014.
[17] A. Amis, R. Prakash, T. Vuong, D. Huynh, Max-min d-cluster formation in wireless ad hoc networks, in: INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, 1, 2000, pp. 32–41, doi:10.1109/INFCOM.2000.832171.
[18] V. Sucasas, A. Radwan, H. Marques, J. Rodriguez, S. Vahid, R. Tafazolli, How much overhead is required for stable group formation in vanets? in: 2014 IEEE 19th International Workshop onComputer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2014, pp. 21–25, doi:10.1109/CAMAD.2014.7033198.
[19] M.R. Brust, H. Frey, S. Rothkugel, Dynamic multi-hop clustering for mobile hybrid wireless networks, in: Proceedings of the 2Nd International Conference on Ubiquitous Information Management and Communication, in: ICUIMC '08, ACM, New York, NY, USA, 2008, pp. 130–135, doi:10.1145/1352793.1352820.
[20] M.R. Brust, H. Frey, S. Rothkugel, Adaptive multi-hop clustering in mobile networks, in: Proceedings of the Fourth International Conference on Mobile Technology, Applications, and Systems and the 1st International Symposium on Computer Human Interaction in Mobile Technology, in: Mobility '07, ACM, New York, NY, USA, 2007, pp. 132–138, doi:10.1145/1378063.1378086.
[21] O. Younis, S. Fahmy, Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks, IEEE Trans. Mobile Comput. 3 (4) (2004) 366–379, doi:10.1109/TMC.2004.41.
[22] W.R. Heinzelman, A. Ch, H. Balakrishnan, in: Energy-Efficient Communication Protocol for Wireless Microsensor Networks, 2000, pp. 3005–3014.
[23] V. Sucasas, A. Radwan, H. Marques, J. Rodriguez, S. Vahid, R. Tafazolli, On the efficiency of merging procedures in hierarchical mobile cooperative networks, IEEE ICC 2014, 2014.
[24] M. Gerla, J.T. chieh Tsai, Multicluster, mobile, multimedia radio network, J. Wireless Networks 1 (1995) 255–265.
[25] Y. Zhang, J.M. Ng, A distributed group mobility adaptive clustering algorithm for mobile ad hoc networks, in: IEEE International Conference on Communications, 2008. ICC '08, 2008, pp. 3161–3165, doi:10.1109/ICC.2008.595.
[26] H. Cheng, J. Cao, X. Wang, S.K. Das, S. Yang, Stability-aware multi-metric clustering in mobile ad hoc networks with group mobility, Wirel. Commun. Mob. Comput. 9 (6) (2009) 759–771, doi:10.1002/wcm.v9:6.
[27] J. Yu, P.H.J. Chong, A survey of clustering schemes for mobile ad hoc networks, Communications Surveys Tutorials, IEEE 7 (1) (2005) 32–48, doi:10.1109/COMST.2005.1423333.
[28] C. Perkins, E. Belding-Royer, S. Das, Ad Hoc On-Demand Distance Vector (AODV) Routing, 2003.
[29] J. Sucec, I. Marsic, Hierarchical routing overhead in mobile ad hoc networks, IEEE Trans. Mobile Comput. 3 (1) (2004) 46–56, doi:10.1109/TMC.2004.1261816.
[30] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, T.-W. Chen, Scalable routing strategies for ad hoc wireless networks, IEEE J. Selected Areas Commun. 17 (8) (1999) 1369–1379, doi:10.1109/49.779920.
[31] T.-C. Hou, T.-J. Tsai, A access-based clustering protocol for multihop wireless ad hoc networks, IEEE J. Selected Areas Commun. 19 (7) (2001) 1201–1210, doi:10.1109/49.932689.
[32] C. Lin, M. Gerla, Adaptive clustering for mobile wireless networks, IEEE J. Selected Areas Commun. 15 (7) (1997) 1265–1275, doi:10.1109/49.622910.
[33] R. Aoudjit, A. Mzoughi, M. Lalam, M. Belkadi, M. Daoui, Load balancing: an approach based on the clustering in ad hoc networks., CIT 17 (2) (2009) 177–184.
[34] M. Buvana, M. Suganthi, K. Muthumayil, Novel architecture for cluster based service discovery and load balancing in mobile ad hoc network, in: 2011 Third International Conference on Advanced Computing (ICoAC), 2011, pp. 292–297, doi:10.1109/ICoAC.2011.6165191.
[35] D. Gavalas, G. Pantziou, C. Konstantopoulos, B. Mamalis, Clustering of mobile ad hoc networks: An adaptive broadcast period approach, in: IEEE International Conference on Communications, 2006. ICC '06., 9, 2006, pp. 4034–4039, doi:10.1109/ICC.2006.255712.
[36] T. Kwon, M. Gerla, V. Varma, M. Barton, T. Hsing, Efficient flooding with passive clustering-an overhead-free selective forward mechanism for ad hoc/sensor networks, Proc. IEEE 91 (8) (2003) 1210–1220, doi:10.1109/JPROC.2003.814920.
[37] R. Krishnan, R. Ramanathan, M. Steenstrup, Optimization algorithms for large self-structuring networks, in: INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, 1, 1999, pp. 71–78, doi:10.1109/INFCOM.1999.749254.
[38] M.R. Brust, S. Rothkugel, Small worlds: Strong clustering in wireless networks, in: 1st International Workshop on Localized Algorithms and Protocols for Wireless Sensor Networks. Santa Fe, NM, USA, 2007.
[39] Y.P. Chen, A.L. Liestman, Approximating minimum size weakly-connected dominating sets for clustering mobile ad hoc networks, in: In Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc, 2002, pp. 165–172.
[40] J. Yu, P.H.J. Chong, 3hbac (3-hop between adjacent clusterheads): a novel non-overlapping clustering algorithm for mobile ad hoc networks, in: 2003 IEEE Pacific Rim Conference on Communications, Computers and signal Processing, 2003. PACRIM, 1, 2003, pp. 318–321, doi:10.1109/PACRIM.2003.1235781.
[41] M. Chatterjee, S.K. Das, D. Turgut, WCA: a weighted clustering algorithm for mobile ad hoc networks, J. Cluster Comput. (Special Issue on Mobile Ad hoc Networks 5 (2001) 193–204.
[42] G.A. Shah, Ö.B. Akan, B. Baykal, Real-time coordination and routing in wireless sensor and actor networks, in: In Proc. 6th Int. Conf. on Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN), Lecture Notes in Computer Science, 2006, pp. 365–383.
[43] T. Ohta, S. Inoue, Y. Kakuda, An adaptive multihop clustering scheme for highly mobile ad hoc networks, in: Autonomous Decentralized Systems, 2003. ISADS 2003. The Sixth International Symposium on, 2003, pp. 293–300, doi:10.1109/ISADS.2003.1193960.
[44] V. Kawadia, P. Kumar, Power control and clustering in ad hoc networks, in: INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies, 1, 2003, pp. 459–469, doi:10.1109/INFCOM.2003.1208697.
[45] A. Fathi, H. Taheri, Enhance topology control protocol(ecec) to conserve energy based clustering in wireless ad hoc networks, in: 2010 Third IEEE International Conference on Computer Science and Information Technology (ICCSIT), 9, 2010, pp. 356–360, doi:10.1109/ICCSIT.2010.5564463.

[46] M. Ahmed, S. Krishnamurthy, R. Katz, S. Dao, An architecture for providing range extension by deploying mobile gateways in ad hoc networks, in: The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2002, 4, 2002, pp. 1660–1664, doi:10.1109/PIMRC.2002.1045461.
[47] A.P. S. Basagni M. Mastrogiovanni, C. Petrioli, Localized Protocols for Ad Hoc Clustering and Backbone Formation: A Performance Comparison, 2006.
[48] J. Wu,et al., Wirelss Network Simulator in Matlab [online].
[49] D.J. Baker, A. Ephremides, radio telecommunication networks, in: ICDCS, IEEE Computer Society, pp. 476–483.
[50] A. Ephremides, J. Wieselthier, D. Baker, A design concept for reliable mobile radio networks with frequency hopping signaling, Proc. IEEE 75 (1) (1987) 56–73, doi:10.1109/PROC.1987.13705.
[51] C. chuan Chiang, M. Gerla, L. Zhang, Forwarding Group Multicast Protocol (FGMP) for Multihop, Mobile Wireless Networks, 1998.
[52] F. Dai, J. Wu, An extended localized algorithm for connected dominating set formation in ad hoc wireless networks, IEEE Trans. Parallel Distrib. Syst. 15 (10) (2004) 908–920, doi:10.1109/TPDS.2004.48.
[53] J. Wu, F. Dai, M. Gao, I. Stojmenovic, On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks, Communications and Networks, Journal of 4 (1) (2002) 59–70, doi:10.1109/JCN.2002.6596934.
[54] S. Bhattacharjee, J. Tripathi, O. Mistry, J. Dattagupta, Distributed algorithm for power aware connected dominating set for efccient routing in mobile ad hoc networks, in: International Symposium on Ad Hoc and Ubiquitous Computing, 2006. ISAUHC '06, 2006, pp. 69–74, doi:10.1109/ISAHUC.2006.4290650.
[55] S. Zahidi, F. Aloul, A. Sagahyroon, W. El-Hajj, Optimizing complex cluster formation in manets using sat/ilp techniques, IEEE Sens. J. 13 (6) (2013) 2400–2412, doi:10.1109/JSEN.2013.2254234.
[56] B. An, S. Papavassiliou, A mobility-based clustering approach to support mobility management and multicast routing in mobile ad-hoc wireless networks, Int. J. Netw. Manag. 11 (6) (2001) 387–395, doi:10.1002/nem.415.
[57] K. Liu, J. Su, J. Zhang, F. Liu, C. Gong, A novel stable cluster protocol for mobile ad hoc networks, in: IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications, 2005. MAPE 2005., vol. 2, 2005, pp. 1328–1332, doi:10.1109/MAPE.2005.1618169.
[58] C.S. Jensen, D. Lin, B.-C. Ooi, Continuous clustering of moving objects, IEEE Trans. Knowl. Data Eng. 19 (9) (2007) 1161–1174, doi:10.1109/TKDE.2007.1054.
[59] S. Leng, Y. Zhang, H.-H. Chen, L. Zhang, K. Liu, A novel k-hop compound metric based clustering scheme for ad hoc wireless networks, Wireless IEEE Trans. Commun. 8 (1) (2009) 367–375, doi:10.1109/T-WC.2009.080186.
[60] Y.-X. Wang, F. Bao, An entropy-based weighted clustering algorithm and its optimization for ad hoc networks, in: Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, 2007. WiMOB 2007., 2007, p. 56, doi:10.1109/WIMOB.2007.4390850.
[61] B. Hassanabadi, C. Shea, L. Zhang, S. Valaee, Clustering in vehicular ad hoc networks using affinity propagation., Ad Hoc Networks 13 (2014) 535–548.
[62] B.J. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (2007) 2007.
[63] Y.-j. Oh, K.-w. Lee, A clustering algorithm based on mobility properties in mobile ad hoc networks, Int. J. Distrib. Sen. Netw. 2015 (2015) 117:117–117:117, doi:10.1155/2015/567269.
[64] O. Banimelhem, M. Mowafi, E. Taqieddin, F. Awad, M. Al Rawabdeh, An efficient clustering approach using genetic algorithm and node mobility in wireless sensor networks, in: Wireless Communications Systems (ISWCS), 2014 11th International Symposium on, 2014, pp. 858–862, doi:10.1109/ISWCS.2014.6933473.
[65] A. Gorrieri, M. Martalò, S. Busanelli, G. Ferrari, Clustering and sensing with decentralized detection in vehicular ad hoc networks, Ad Hoc Networks 36 (2016) 450–464, doi:10.1016/j.adhoc.2015.05.019.
[66] S. Kukliński, G. Wolny, Density based clustering algorithm for vanets, in: Fifth International Conference on Testbeds and Research Infrastructures for the Development of Networks Communities and Workshops, 2009. TridentCom 2009, 2009, pp. 1–6, doi:10.1109/TRIDENTCOM.2009.4976256.
[67] Z. Rawshdeh, S. Mahmud, Toward strongley connected clustering structure in vehicular ad hoc networks, in: Vehicular Technology Conference Fall (VTC 2009-Fall), 2009 IEEE 70th, 2009, pp. 1–5, doi:10.1109/VETECF.2009.5378799.
[68] C. Caballero-Gil, P. Caballero-Gil, J. Molina-Gil, Self-organized clustering architecture for vehicular ad hoc networks, Inte. J. Distrib. Sens. Networks vol. 2015 (ID 384869) (2015) 5851–5857, doi:10.1155/2015/384869.
[69] K. Abboud, W. Zhuang, Stochastic modeling of single-hop cluster stability in vehicular ad hoc networks, IEEE Transactions on Vehicular Technology 65 (1) (2016) 226–240, doi:10.1109/TVT.2015.2396298.
[70] P. Basu, N. Khan, T.D.C. Little, A mobility based metric for clustering in mobile ad hoc networks, in: 2001 International Conference on Distributed Computing Systems Workshop, 2001, pp. 413–418, doi:10.1109/CDCS.2001.918738.
[71] Z.Z. Ni Minming, Z. Dongmei, Doppler shift based stable clustering scheme for mobile ad hoc networks, High Technol. Lett. 17 (4) (2011) 414–418.
[72] M. Ni, Z. Zhong, D. Zhao, Mpbc: A mobility prediction-based clustering scheme for ad hoc networks, IEEE Trans. Vehic. Technol. 60 (9) (2011) 4549–4559, doi:10.1109/TVT.2011.2172473.
[73] A. Zajic, Estimation of mobile velocities and direction of movement in mobile-to-mobile wireless fading channels, IEEE Trans. Vehic. Technol. 61 (1) (2012) 130–139, doi:10.1109/TVT.2011.2175410.
[74] B. Gu, X. Hong, Mobility identification and clustering in sparse mobile networks, in: Military Communications Conference, 2009. MILCOM 2009. IEEE, 2009, pp. 1–7, doi:10.1109/MILCOM.2009.5379893.
[75] V. Sucasas, A. Radwan, H. Marques, J. Rodriguez, R. Tafazolli, Moblist: A signal strength based clustering algorithm for ordered mobile scenarios, in: Globecom Workshops (GC Wkshps), 2012 IEEE, 2012, pp. 380–385, doi:10.1109/GLOCOMW.2012.6477601.
[76] A. McDonald, T. Znati, Design and performance of a distributed dynamic clustering algorithm for ad-hoc networks, in: Simulation Symposium, 2001. Proceedings. 34th Annual, 2001, pp. 27–35, doi:10.1109/SIMSYM.2001.922111.
[77] P. Bellavista, E. Magistretti, How node mobility affects k-hop cluster quality in mobile ad hoc networks: a quantitative evaluation., in: ISCC, IEEE, 2008, pp. 750–756.
[78] Y. Li, P. Wang, A load balance k-hop clustering algorithm for ad hoc networks, in: Fifth International Conference on Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09, 2009, pp. 1–4, doi:10.1109/WICOM.2009.5301694.
[79] L. Conceiçao, M. Curado, Onto scalable wireless ad hoc networks: Adaptive and location-aware clustering, Ad Hoc Networks 11 (8) (2013) 2484–2499.
[80] E. Dror, C. Avin, Z. Lotker, Fast randomized algorithm for hierarchical clustering in vehicular ad-hoc networks, in: Ad Hoc Networking Workshop (Med-Hoc-Net), 2011 The 10th IFIP Annual Mediterranean, 2011, pp. 1–8, doi:10.1109/Med-Hoc-Net.2011.5970488.
[81] G. Angione, P. Bellavista, A. Corradi, E. Magistretti, A k-hop clustering protocol for dense mobile ad-hoc networks, in: 26th IEEE International Conference on Distributed Computing Systems Workshops, 2006. ICDCS Workshops 2006, 2006, p. 10, doi:10.1109/ICDCSW.2006.7.10–10
[82] A. Amis, R. Prakash, Load-balancing clusters in wireless ad hoc networks, in: Application-Specific Systems and Software Engineering Technology, 2000. Proceedings. 3rd IEEE Symposium on, 2000, pp. 25–32, doi:10.1109/ASSET.2000.888028.
[83] N. Aydin, F. Nait-Abdesselam, V. Pryyma, D. Turgut, Overlapping clusters algorithm in ad hoc networks, in: Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE, 2010, pp. 1–5, doi:10.1109/GLOCOM.2010.5683454.
[84] H. Cheng, S. Yang, Genetic algorithms with elitism-based immigrants for dynamic load balanced clustering problem in mobile ad hoc networks, in: 2011 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE), 2011, pp. 1–7, doi:10.1109/CIDUE.2011.5948486.
[85] W.B. Heinzelman, A.P. Chandrakasan, H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, IEEE Transactions on Wireless Communications 1 (4) (2002) 660–670, doi:10.1109/TWC.2002.804190.
[86] D. Zhixiang, Q. Bensheng, Three-layered routing protocol for wsn based on leach algorithm, in: IET Conference on Wireless, Mobile and Sensor Networks, 2007. (CCWMSN07), 2007, pp. 72–75, doi:10.1049/cp:20070086.
[87] J.S. Lee, T.Y. Kao, An improved three-layer low-energy adaptive clustering hierarchy for wireless sensor networks, IEEE Internet Things J. PP (99) (2016) 1, doi:10.1109/JIOT.2016.2530682.
[88] Z. Abrams, A. Goel, S. Plotkin, Set k-cover algorithms for energy efficient monitoring in wireless sensor networks, in: Third International Symposium on Information Processing in Sensor Networks, 2004. IPSN 2004, 2004, pp. 424–432, doi:10.1109/IPSN.2004.1307364.
[89] M. Alam, J. Rodriguez, A dual head clustering mechanism for energy efficient wsns, in: Mobile Lightweight Wireless Systems, in: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 45, Springer, Berlin, Heidelberg, 2010, pp. 380–387, doi:10.1007/978-3-642-16644-0_33.
[90] D. Jia, H. Zhu, S. Zou, P. Hu, Dynamic cluster head selection method for wireless sensor network, IEEE Sens. J. 16 (8) (2016) 2746–2754, doi:10.1109/JSEN.2015.2512322.
[91] K. Manousakis, J. Baras, Clustering for transmission range control and connectivity assurance for self configured ad hoc networks, in: Military Communications Conference, 2003. MILCOM '03. 2003 IEEE, 2, 2003, pp. 1042–1047, doi:10.1109/MILCOM.2003.1290311.
[92] S. Ghiasi, A. Srivastava, X. Yang, Sarrafzadeh, Optimal energy aware clustering in sensor networks, Sensors 2 (7) (2002) 258–269.
[93] S. Bandyopadhyay, E. Coyle, An energy efficient hierarchical clustering algorithm for wireless sensor networks, in: INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies, vol. 3, 2003, pp. 1713–1723, doi:10.1109/INFCOM.2003.1209194.
[94] T. Parker, J. McEachen, Cluster head selection in wireless mobile ad hoc networks using spectral graph theory techniques, in: HICSS, 2016, 2016, pp. 5851–5857, doi:10.1109/HICSS.2016.724.
[95] I. Sohn, J.H. Lee, S.H. Lee, Low-energy adaptive clustering hierarchy using affinity propagation for wireless sensor networks, IEEE Commun. Lett. 20 (3) (2016) 558–561, doi:10.1109/LCOMM.2016.2517017.
[96] Y.-C. Chen, C.-Y. Wen, Distributed clustering with directional antennas for wireless sensor networks, IEEE Sens. J. 13 (6) (2013) 2166–2180, doi:10.1109/JSEN.2013.2249659.
[97] W. dong Yang, G. zhao Zhang, A weight-based clustering algorithm for mobile ad hoc network, Int. Conf. Wireless Mobile Commun. 0 (2007) 3. http://doi.ieeecomputersociety.org/10.1109/ICWMC.2007.14.

[98] M.R. Brust, A. Andronache, S. Rothkugel, Waca: A hierarchical weighted clustering algorithm optimized for mobile hybrid networks, in: Third International Conference on Wireless and Mobile Communications, 2007. ICWMC '07, 2007, p. 23, doi:10.1109/ICWMC.2007.93.

[99] Y.-X. Wang, F. Bao, An entropy-based weighted clustering algorithm and its optimization for ad hoc networks, in: Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, 2007. WiMOB 2007, 2007, p. 56, doi:10.1109/WIMOB.2007.4390850.

[100] C.-W. Wu, T.-C. Chiang, L.-C. Fu, An ant colony optimization algorithm for multi-objective clustering in mobile ad hoc networks, in: Evolutionary Computation (CEC), 2014 IEEE Congress on, 2014, pp. 2963–2968, doi:10.1109/CEC.2014.6900458.

[101] T.-C. Hou, T.-J. Tsai, A access-based clustering protocol for multihop wireless ad hoc networks, IEEE J. Selected Areas Commun. 19 (7) (2001) 1201–1210, doi:10.1109/49.932689.

[102] C. Crespo, J. Alonso-Zarate, L. Alonso, C. Verikoukis, Distributed point coordination function for wireless ad hoc networks, in: Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th, 2009, pp. 1–5, doi:10.1109/VETECS.2009.5073505.

[103] T. Chen, H. Zhang, M. Matinmikko, M.D. Katz, Cogmesh: Cognitive wireless mesh networks, in: GLOBECOM Workshops, 2008 IEEE, 2008, pp. 1–6, doi:10.1109/GLOCOMW.2008.ECP.37.

[104] D. Xie, Q. Zhou, X. You, B. Li, X. Yuan, A novel energy-efficient cluster formation strategy: From the perspective of cluster members, Communications Letters, IEEE 17 (11) (2013) 2044–2047, doi:10.1109/LCOMM.2013.100813.131109.

[105] Y. Harikrishnan, J. He, Clustering algorithm based on minimal path loss ratio for vehicular communication, in: 2013 International Conference on Computing, Networking and Communications (ICNC), 2013, pp. 745–749, doi:10.1109/ICCNC.2013.6504181.

[106] R. Rajagopalan, P.K. Varshney, Data aggregation techniques in sensor networks: A survey, IEEE Commun. Surv. Tutor. 8 (2006) 48–63.

[107] P. Jesus, C. Baquero, P.S. Almeida, A survey of distributed data aggregation algorithms, IEEE Commun. Surv. Tutor. 17 (1) (2015) 381–404, doi:10.1109/COMST.2014.2354398.

[108] J. Kulik, W. Heinzelman, H. Balakrishnan, Negotiation-based protocols for disseminating information in wireless sensor networks, Wirel. Netw. 8 (2/3) (2002) 169–185, doi:10.1023/A:1013715909417.

[109] C. Intanagonwiwat, R. Govindan, D. Estrin, Directed diffusion: A scalable and robust communication paradigm for sensor networks, in: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, in: MobiCom '00, ACM, New York, NY, USA, 2000, pp. 56–67, doi:10.1145/345910.345920.

[110] S. Lindsey, C. Raghavendra, K.M. Sivalingam, Data gathering algorithms in sensor networks using energy metrics, IEEE Trans. Parallel Distrib. Syst. 13 (9) (2002) 924–935, doi:10.1109/TPDS.2002.1036066.

[111] M. Ding, X. Cheng, G. Xue, Aggregation tree construction in sensor networks, in: Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th, 4, 2003, pp. 2168–2172, doi:10.1109/VETECF.2003.1285913.

[112] K. Vaidhyanathan, Data aggregation techniques in sensor networks, Technical Report, OSU-CISRC-11/04-TR60, Ohio State University, 2004.

[113] S. Chatterjea, P. Havinga, A dynamic data aggregation scheme for wireless sensor networks, ProRISC 2003, 14th Workshop on Circuits, Systems and Signal Processing, 2003.

[114] L. Atzori, A. Iera, G. Morabito, The internet of things: A survey, Comput. Netw. 54 (15) (2010) 2787–2805, doi:10.1016/j.comnet.2010.05.010.

[115] A.P. Bianzino, C. Chaudet, D. Rossi, J.-L. Rougier, A survey of green networking research., IEEE Commun. Surv. Tutor. 14 (1) (2012) 3–20.

[116] M. Taneja, A framework for power saving in iot networks, in: 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI, 2014, pp. 369–375, doi:10.1109/ICACCI.2014.6968211.

[117] D. Malak, H.S. Dhillon, J.G. Andrews, Optimizing data aggregation for uplink machine-to-machine communication networks, IEEE Trans. Commun. 64 (3) (2016) 1274–1290, doi:10.1109/TCOMM.2016.2517073.

[118] S.Q. Ren, K.M.M. Aung, J.S. Park, A privacy enhanced data aggregation model., in: CIT, IEEE Computer Society, 2010, pp. 985–990.

[119] Y. Liu, X. Gong, C. Xing, A novel trust-based secure data aggregation for internet of things, in: 2014 Ninthth International Conference on Computer Science Education (ICCSE), 2014, pp. 435–439, doi:10.1109/ICCSE.2014.6926499.

[120] T. Limmer, F. Dressler, Improving the performance of intrusion detection using dialog-based payload aggregation, in: 2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2011, pp. 822–827, doi:10.1109/INFCOMW.2011.5928926.

[121] F. Dressler, G. Munz, Flexible flow aggregation for adaptive network monitoring, in: Proceedings. 2006 31st IEEE Conference on Local Computer Networks, 2006, pp. 702–709, doi:10.1109/LCN.2006.322180.

[122] J. Sahoo, S. Cherkaoui, A. Hafid, A novel vehicular sensing framework for smart cities, in: 39th Annual IEEE Conference on Local Computer Networks, 2014, pp. 490–493, doi:10.1109/LCN.2014.6925824.

[123] S. Ucar, S.C. Ergen, O. Ozkasap, Vesca: Vehicular stable cluster-based data aggregation, in: 2014 International Conference on Connected Vehicles and Expo (ICCVE), 2014, pp. 1080–1085, doi:10.1109/ICCVE.2014.7297517.

[124] W. Zhou, D. Saha, S. Rangarajan, A system architecture to aggregate video surveillance data in smart cities, in: 2015 IEEE Global Communications Conference (GLOBECOM), 2015, pp. 1–7, doi:10.1109/GLOCOM.2015.7417602.

[125] Q. Mahmoud, Cognitive Networks: Towards Self-Aware Networks, Wiley-Interscience, 2007.

[126] J. Surowiecki, The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations, Doubleday, 2004.

[127] V. Sucasas, A. Radwan, H. Marques, J. Rodriguez, S. Vahid, R. Tafazolli, A cognitive approach for stable cooperative group formation in mobile environments, IEEE ICC 2014 (2014).

[128] Y. Jin, Y. Meng, Morphogenetic robotics: An emerging new field in developmental robotics, IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. 41 (2) (2011) 145–160, doi:10.1109/TSMCC.2010.2057424.

[129] S.M. Mousavi, Feature selection for mobility pattern recognition in mobile ad-hoc networks, in: WiCOM, 2011, 2011, pp. 1–4, doi:10.1109/wicom.2011.6040418.

[130] W. Ma, Y. Fang, P. Lin, Mobility management strategy based on user mobility patterns in wireless networks, IEEE Trans. Vehic. Technol. 56 (1) (2007) 322–330, doi:10.1109/TVT.2006.883743.

[131] K. Mershad, H. Artail, M. Gerla, We can deliver messages to far vehicles, IEEE Trans. Intell. Transp. Syst. 13 (3) (2012) 1099–1115, doi:10.1109/TITS.2012.2183124.

[132] J. Zhao, C. Qiao, R. Sudhaakar, S. Yoon, Improve efficiency and reliability in single-hop wsns with transmit-only nodes, IEEE Trans. Parallel Distrib. Syst. 24 (3) (2013) 520–534, doi:10.1109/TPDS.2012.148.

[133] M. Munir, F. Filali, Increasing connectivity in wireless sensor-actuator networks using dynamic actuator cooperation, in: Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE, 2008, pp. 203–207, doi:10.1109/VETECS.2008.54.

[134] R. Lent, E.-H. Ngai, Lightweight clustering in wireless sensor-actuator networks on obstructed environments, in: Third International Symposium on Wireless Pervasive Computing, 2008. ISWPC 2008, 2008, pp. 26–32, doi:10.1109/ISWPC.2008.4556159.

Victor Sucasas received his degree in Telecommunications Engineering from the University of Vigo (Spain), in 2008, with the specialization of Telematics. His studies include a 6 month stage at the K.U.Leuven (Belgium) where he did his Master Thesis in collaboration with the COSIC group. In 2008, he started working as a collaborator with the Signal Theory Research Group at the University of Vigo where he was involved in projects with ESA and AENA. In June 2011 he joined the 4TELL group at Instituto de Telecomunicações, Aveiro, as a Marie Curie Fellow. He is also a Ph.D. student at ICS in the University of Surrey, U.K. His main research interests include wireless cooperative communications, ad hoc networks, network security and privacy preserving systems.

Ayman Radwan received his Master of Applied Science (MASc) from Carleton University (Canada) and Ph.D. from Queen's University (Canada), in 2003 and 2009, respectively. In January 2010, he joined Instituto de Telecomunicações. Since joining, Dr. Radwan has been intensively active in European projects. He acted as the Technical Manager of the FP7 C2POWER project and the Project Manager of the CELTIC Green-T project. Dr. Radwan is an active IEEE member for years and, as a representative of the 4TELL Group, is a founding member of the IEEE Communications Society Technical Subcommittee on Green Communications and Computing. The focus of his research interests includes next generation mobile networks (5G), green communications, radio resource management, quality of service provision, Internet of Things and Smart City.

Hugo Marques received his BSc. and MSc. degrees in 1998 and 2002, respectively, from the Department of Electronics and Telecommunications, University of Aveiro, Portugal. He is a Senior Researcher at Instituto de Telecomunicações in Aveiro, Portugal, where he develops extensive research in the fields of cognitive radio, seamless and vertical handovers, network security, highly resilient emergency networks, secure localization on Wireless networks, energy efficient wireless networks, low delay, high resilient surveillance applications and hybrid delivery of 3D content via DVB and peer-to-peer networks. He is currently a Ph.D. candidate in the Centre for Communication Systems Research (CCSR) at the University of Surrey.

Jonathan Rodriguez received his MEng (Hons) and Ph.D in Electronic and Electrical Engineering from the University of Surrey (UK), in 1998 and 2004. He became a researcher at the Instituto de Telecomunicações – Portugal in 2005, establishing the 4TELL Research Group (http://www.av.it.pt/4TELL/). He has served as project coordinator on the Eureka-LOOP, FP7-C2POWER, and as technical manager on FP7-COGEU and FP7-SALUS. In 2009 he became an Invited Assistant Professor at the University of Aveiro (Portugal), and Associate in 2015. He is author of more than 300 scientific works, including 8 book editorials. He is a Senior Member of IEEE and Chartered Engineer (2013), and Fellow of the IET (2015).

Seiamak Vahid received his B.Sc. (hon.), M.Sc., and Ph.D. degrees in electrical and electronic engineering from Kings College, University of London, and University of Surrey (UK), in 1985, 1987, and 2000, respectively. From 1988 to 2000, he was a research fellow in the Institute for Communication Systems, University of Surrey. From 2001 to 2010, he was with the GSM Networks division, Motorola UK, as team leader and later as 3G/LTE system architect and product security specialist. Since April 2010 he has been back at ICS as senior research fellow and his research interests include the areas of cognitive radio, communication systems modeling, multiple access, resource management, routing and optimization techniques and their applications to cellular and ad hoc wireless communication network.

Rahim Tafazolli is the director of Institute of Communication Systems (ICS, formerly known as CCSR), Faculty of Engineering and Physical Sciences, The University of Surrey in the UK. ICS is the largest academic communications research center in Europe with extensive activities with academia and industry in the UK, Europe and the rest of the world. He has published more than 500 research papers in refereed journals, international conferences and as invited speaker. He currently has more than 15 patents in the field of mobile communications. He is the editor of two books on "Technologies for Wireless Future" published by Wiley's Vol. 1 in 2004 and Vol. 2 2006. He is currently the Chairman of EU NetWorks Technology Platform Expert Group, board member of the UK Future Internet Strategy Group (UK-FISG) and coordinator of the IoT-i coordination action project. He is a Fellow of IET and WWRF (Wireless World Research Forum).