Computer Networks 81 (2015) 220–244

Contents lists available at ScienceDirect

Computer Networks

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

Time-optimized user grouping in Location Based Services

Christos Anagnostopoulos [a,*,1], Stathes Hadjiefthymiades [b], Kostas Kolomvatsos [c]

[a] Department of Informatics, Ionian University, 49100, Greece
[b] Department of Informatics and Telecommunications, University of Athens, 15784, Greece
[c] Department of Computer Science, University of Thessaly, 35100, Greece

ARTICLE INFO

Article history:
Received 27 April 2014
Received in revised form 25 January 2015
Accepted 6 February 2015
Available online 28 February 2015

Keywords:
Group computing
Location-Based Services
Optimal Stopping Theory

ABSTRACT

We focus on Location Based Services (LBSs) which deliver information to groups of mobile users based on their spatial context. The existence (and non-trivial lifetime) of groups of mobile users can simplify the operation of LBS and reduce network overhead (due to location updates and application content flow). We propose an incremental group formation algorithm and an optimally scheduled, adaptive group validation mechanism which detects and further exploits spontaneous formation of groups of mobile users. The advantages of application simplification and network load reduction are pursued. Through the proposed mechanism the LBS server (back-end system) monitors representatives of formed groups, i.e., group leader, and, in turn, disseminates location-dependent application content to group members. We first introduce an incremental group formation algorithm for group partitioning and identification of group leaders. We elaborate on the mechanism which adopts the Optimal Stopping Theory in order to assess the group persistence through evaluation of compactness and coherency metrics. We compare the performance of the proposed scheme with that of existing mechanisms for moving objects clustering and quantify the benefits stemming for its adoption.

1. Introduction

We focus on the Location Based Services (LBS), which deliver information to groups of mobile users based on their surroundings, e.g., location information, nearby objects, spatially characterized content. In a typical scenario, mobile clients equipped with Global Positioning System (GPS) devices or other positioning technology send their locations to a central LBS server; hereafter referred to as the *system*, as shown in Fig. 1(a). The system collects such location information, establishes a connection with each mobile terminal and delivers (possibly personalized) content, e.g., advertisements to mobile users upon entrance to specific areas, like shopping malls, warnings when traffic conditions change, or tourist information when users approach a certain location [1,2]. However, as the number of users increases and they continuously change their position, the system and network load for location-based content delivery becomes quite significant [3].

Mobile users often form groups in daily life. The partition of moving objects into groups (clusters) has attracted increasing attention in the recent past [4–6]. A *group* is formally defined by a *Group Leader* (GL) and nodes that are within the GL's communication range. Such nodes are, hereafter, referred to as *members* of the group. Moving objects clustering results to groups of objects, which are not only close to each other at the current time but also

* Corresponding author at: School of Computing Science, University of Glasgow, United Kingdom.
E-mail addresses: christos@ionio.gr, christos.anagnostopoulos@glasgow.ac.uk (C. Anagnostopoulos), shadj@di.uoa.gr (S. Hadjiefthymiades), kolomvatsos@cs.uth.gr (K. Kolomvatsos).
[1] Dr C. Anagnostopoulos is a Postdoctoral Fellow at the School of Computing Science, University of Glasgow, United Kingdom.

http://dx.doi.org/10.1016/j.comnet.2015.02.017
1389-1286/

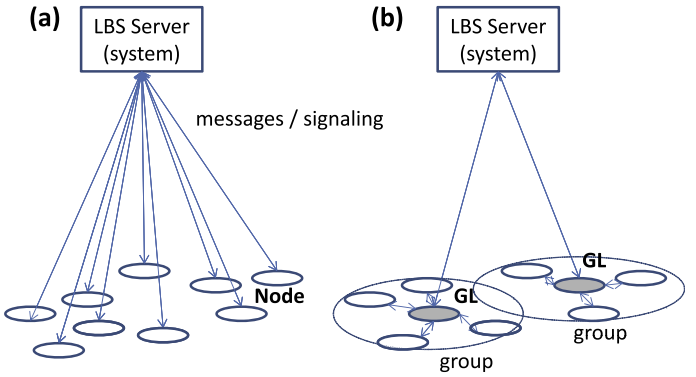


Fig. 1. (a) Typical scenario of location-dependent content and signalling between LBS server (system) and moving objects (nodes), and (b) Group-oriented communication among LBS server, group leaders (GLs) and nodes.

likely to move together for a certain time horizon. In Fig. 2, we can see an example of three group examples of moving objects Q_1, Q_2, Q_3 and their evolution over time starting from their creation time instance t_0. Evidently, the movement of objects that belong to a certain group is highly correlated.

The work presented in this paper deals with *time-optimized group monitoring*. Our motivation is to exploit moving object groups for minimizing location updates from mobile users to the system and application related messages from the system to mobile users as shown in Fig. 1(b). Specifically, the concept behind our model is that the system, instead of monitoring the location of all individuals (and thus transmitting location-dependent content to them), it (i) identifies persisting groups of users, (ii) discontinues monitoring individuals, and (iii) monitors only the representative (a single member) of a group i.e., GL. Based on this approach, the system pushes application content only to the GL and in turn the GL passes such content to group members. In terms of signalling, after the appointment of a member as GL, the GL notifies group members to stop reporting their locations to the system. Hence, the use of resources in the underlying network is highly rationalized and system's computation overhead is significantly reduced. As a result new group-oriented services can be introduced in back-end systems and dispatched easily through the supporting network.

Our model tries to handle problems arisen when groups are moving. A moving group should be observed over time since some members with different speeds or diverging behaviour over time may render the group not so compact, as shown in Fig. 2. In such case the system periodically checks whether the initial group retains its 'structure', i.e., the number and identity of members persists, in order to monitor its members through the GL. However, the large number of continuous location changes of users increases the system load (e.g., high frequency of location updates) resulting to increased network overhead. The decision of the system whether to communicate only with GL of an already identified group or with all members of the group should be validated. This decision depends significantly on the compactness and coherency of each group (to be introduced in the following paragraphs). It is deemed appropriate that the system be certain enough to decide whether group persists or not in a specific time horizon. This motivated us to define and propose a Group Validation System (GVS) to support a group-oriented system on taking such decisions.

Consider a formed group Q at time instance t. The GVS continuously monitors the persistence behaviour of Q before deciding to communicate only with the GL. Obviously, if the GVS delays on making a decision to monitor only the GL of the group other than monitoring all members then we obtain increased load of network and computational overhead due to (i) continuous location updates from all group members and (ii) delivery of location-dependent content from the system to all members. On the other hand as long as the GVS delays on making such decision, it becomes more confident on the persistence of the group. The problem for the GVS is to determine a time instance $t_* > t$ at which it stops monitoring and communicating with all group members. The time

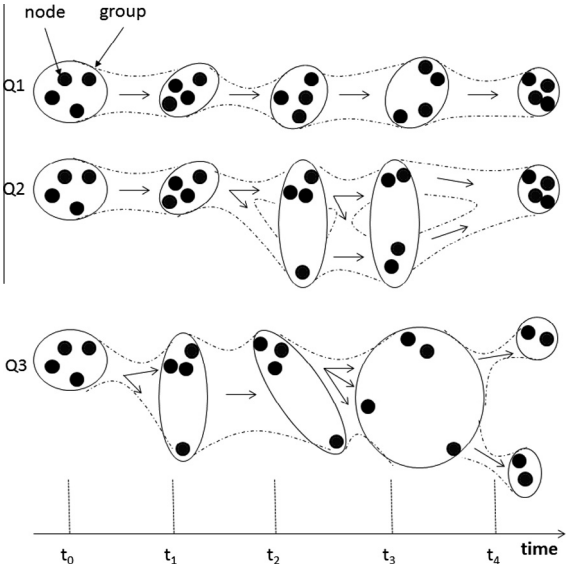


Fig. 2. Groups of moving objects (nodes), Q_1, Q_2, Q_3, along the time span.

instance t_* is associated with a high belief that the group will be persistent (i.e., consisting of the same members as initially structured) for certain time instances $t' > t_*$. Once the GVS decides to communicate only with the GL instead of all group members then we achieve network load reduction, however, at the risk that the group might be no longer persistent at some time instance $t' > t_*$. We treat this optimally scheduled decision problem as an *optimal stopping problem* treated through the Optimal Stopping Theory (OST) [12] (see Section 5.1). The optimal stopping time t_* in our problem is the milestone when the GVS stops monitoring all members of the group and then communicates only with the GL. The determination of t_* is achieved by adopting the fundamental axioms of OST since t_* is a random variable. Specifically, the movement of the members of the groups is definitely not deterministic. Hence the corresponding persistence behaviour of the group is quantified in this paper through the introduced group cohesion metric in Section 3.2. The GVS has to determine an optimal stopping rule over the random representation of the group persistence in a stochastic manner in order to achieve optimality. However, a deterministic stopping rule which refers to a decision of the GVS at a fixed time instance could be another alternative solution to our problem. As it will be illustrated, OST outperforms all deterministic stopping rules.

The contribution of this paper is summarized as follows: firstly, we propose an incremental Group Formation Algorithm (GFA) as a moving objects clustering method and introduce the group persistence concept and group cohesion metric. In addition, we introduce a broader GVS, which exploits GFA and employs sequential decision making on the persistence of group based on OST. Moreover, we propose an adaptation mechanism for GVS to balance the system load. We provide also performance and comparative assessments of GVS with other approaches found in the literature [13] and compare the proposed optimal stopping rule with deterministic stopping rules.

The differences of our work compared to other relevant research efforts are as follows: we introduce an incremental GFA in order to remedy the basic drawbacks of static and continuous GFAs. The proposed GFA (a) makes no assumptions about the number of groups and number of moving objects that can be grouped, (b) requires no knowledge on motion patterns of moving objects, and (c) conveys a physical meaning of a formed group in terms of intra-group communication capability (between members of the group). Moreover, the proposed incremental GFA exhibits a good performance compared to other efforts. For instance, the proposed GFA algorithm outperforms the algorithm in [14] as shown in Section 6.4. It should be noted that the remaining research efforts in the field assume a fixed number of groups that can be formed through their clustering mechanisms. As a consequence, the derived clusters are not associated with the physical meaning of a group, which implies that all members of a group should be able to communicate with each other (a fundamental attribute of a group-oriented system). Furthermore, we exploit the incremental GFA in an optimal way to decrease the system load. It should be noted also that our scheme does not depend on a pre-defined time horizon as considered in [14], since GVS has to autonomously determine when to validate a group. Finally, we introduce the cohesion metric with a very specific physical meaning, which generalizes the analogous metric (referred to as group validity metric in [14]).

The paper is organized as follows: In Section 2, we discuss the related research efforts with our approach. Section 3 introduces the group cohesion metric and incremental GFA. Section 4 defines the group validation problem, while Section 5 presents the time-optimized GVS. Section 6 provides performance and comparative assessment of the proposed scheme. In Section 7, we provide a discussion on the strengths and weaknesses of the proposed model while in Section 8, we present some application domains where our model fits well. Finally, Section 9 concludes the paper.

2. Related work

There are several approaches on how to maintain clustering information of groups of moving objects in presence of mobility. Clustering re-evaluation (re-clustering) due to node mobility can be performed (i) periodically by invoking the Group Formation Algorithm (GFA), or (ii) continuously by collecting specific information on groups' structure. A straightforward approach of grouping a large set of moving objects is to perform a static GFA periodically. Research has also focused on continuous GFA. A continuous GFA captures each group change as it occurs during a continuous observation process.

A comprehensive survey for static GFAs is provided in [15]. Specifically, many group formation techniques have been proposed for static datasets. They can be classified into the partitioning [22], hierarchical [18,24], density-based [23], grid-based [25,16], model-based [17] and soft-computing-based (by adopting genetic algorithms e.g., [30–32], and fuzzy set theory e.g., [28]) GFAs. There are also a few approaches [19,21,26] on grouping objects in spatial networks, where these GFAs avoid computing distances between every pair of objects by exploiting the properties of the network. However, all these solutions assume a static dataset. As mentioned before, a straightforward extension of these GFAs to moving objects by periodical re-evaluation is inefficient. Besides, the model in [20] studies the problem of mining distance-based outliers in spatial networks, but it is only a byproduct of group formation. In the static GFA model in [33] every snapshot of the formed groups is determined by the previous one, thus, creating a temporal dependency between snapshots. The model in [33] incurs large communication and computational cost due to frequent updates. Approaches based on static GFAs produce high quality results in terms of clustering information as they continuously follow the location of every object. On the other hand, static GFA incurs large communication (between moving objects and back-end system) and computational cost due to frequent updates and a high rate of re-clustering. It is worth noting that static GFAs are usually invoked for

detecting moving objects patterns from movement trajectories, like [38] instead of periodically invoked for group formation in group-oriented environments.

Approaches based on continuous GFAs requires continuous transactions between moving objects and the system. The continuous GFA in [34] applies micro-clustering [35] to moving objects and dynamically maintains bounded boxes of formed groups to detect changes. The number of 'maintenance' events (events which indicate whether a member is located outside a bounded box) involved is prohibitively large. Given a set of $|V|$ objects, maintenance events can occur up to $O(|V|)$ times [5]. There are continuous GFAs that capture and analyse relations among groups (after group formation) in order to obtain updated clustering information. The model in [36] compares the structure of groups that appear in consecutive snapshots. The comparison cost can be very high especially when successive groups share a large number of common members. In a quite similar way, the scheme in [37] applies a GFA at multiple time instances to detect group changes. Existing approaches, e.g., [4,5,36], require the moving objects to report every location update, which is prohibitively expensive in terms of network overhead and battery power. In addition, certain approaches such as [4] are limited to the case of linearly moving objects. Clustering moving objects is also used for spatio-temporal selectivity estimation [27]. The model in [29] represents the uncertainty of moving objects by spatial density functions associated with the likelihood that a certain object is located at a certain position. The scheme in [13] is based on a threshold-based *K*-means monitoring method (TKM). TKM determines specific spatial boundaries (thresholds) for each moving object, which are user-defined. Location update is issued to the system every time a moving object crosses its threshold. In this case, *K*-means GFA is re-evaluated and new thresholds are disseminated from the system to moving objects.

Both approaches (static and continuous GFA) have some drawbacks when adopted for group-oriented systems. Specifically, a static GFA has to be invoked periodically. If the period is short then this approach is overly expensive. If the period is long then the clustering information (e.g., structure of groups) will be of low accuracy (outdated). A continuous GFA requires the moving objects to report every location update, which is prohibitively expensive in terms of network overhead and battery power. In addition, the GFAs found in the literature assume a predefined number of moving objects for performing clustering. However, this is typically not the case in real life conditions. Moving objects can leave/enter a certain geographical area, thus, a typical GFA has to re-determine the already formed groups in order to incorporate new moving objects. Moreover, all the above mentioned GFAs assume a fixed number of groups that can be formed through their clustering mechanisms. As a consequence, the derived clusters are not associated with the physical meaning of a group, which implies that all members of a group should be able to communicate with each other; this is a fundamental attribute of a group-oriented system. Finally, the model in [14] adopts *K*-means static GFA and proposes a group monitoring and validation scheme running for fixed time horizon.

3. Group formation and validation

We distinguish two phases for group management in a mobile network: the *group formation* phase and the *group validation* phase. The former phase refers to partitioning a set of moving objects (nodes) into groups. For each group, a node acts as GL. It should be noted that we study a scenario where nodes act altruistic in the sense that they participate in the discussed scheme with the ultimate goal of gaining access to messages through the GL and, thus, save resources. Each node gains from the group and, if necessary, accepts to act as a GL as in the case where its resources are eliminated, it will receive its messages through the next GL. Hence, in that sense, every node is eager to play the role of the GL when it is necessary. Nonetheless, the altruistic assumption of a GL appointment implies also that a node is capable to be operational as GL (e.g., it has the adequate energy resources). In Section 5.5 we discuss two mechanisms for handling the situation at which an appointed GL is no longer operational and thus the system has to proceed with a GL substitution process. In our future research agenda, the creation of incentives e.g., gain of credits on access or another intelligent scheme for someone is acting as GL is in the first places. A formed group is characterized by its coherency and compactness:

- **Coherency** denotes the rate of membership change (i.e., how often a member leaves/joins a group).
- **Compactness** indicates that group members should be as close to each other as possible [40].

The validation phase involves the assessment of compactness and coherency of a group over time and the conclusion on the **persistence** of the group. In the validation phase the system validates whether a group persists for a certain time horizon. Once the system concludes on the persistence of the group then the group is treated as candidate for fragmentation. In this work, we assume that a group is fragmented if at least one member is beyond the communication range of GL. If group turns fragmented then the formation phase starts again.

The Finite State Machine (FSM) for a group is shown in Fig. 3. Initially, in the Group Formation state (GF), a group is formed. Then the group transits to the Optimally Scheduled state (OS), in which its persistence is assessed through group monitoring. Then the group transits to the Candidate for Fragmentation state (CF), in which it is treated as a unit through GL until it turns fragmented (and transits to the GF state).

Before discussing the problem formulation, we introduce the incremental GFA and the group cohesion metric.

3.1. Group formation algorithm

The basic properties of a GFA are (i) every member has a unique GL (dominance property) and (ii) no two GLs can be neighbours (independence property) [39]. GFA knows the exact position of each node. This information can be delivered directly by a node to the system that executes the

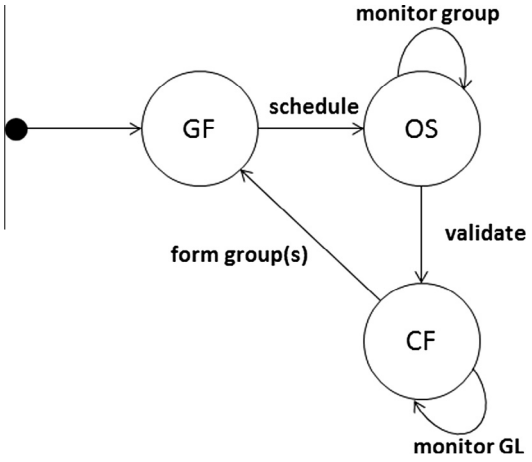


Fig. 3. The group finite state machine; Group Formation state (GF), Optimally Scheduled state (OS), Candidate for Fragmentation state (CF).

GFA. Note that, once groups do not turn fragmented during a certain time interval, there is no need for re-clustering.

We model a mobile network as an undirected graph (V, E), where V is the set of moving nodes and there is an edge $\{u, v\} \in E$ iff $u \in V$ and $v \in V$ can mutually receive each other's transmitted information. This implies that all the links between the nodes are bidirectional. In this case, nodes u and v are neighbours. Let R be the communication range of each node. A node u has as coordinates (longitude, latitude) the point $\mathbf{p}_u$. The node v is neighbour with u, i.e., $\{u, v\} \in E$, iff $\|\mathbf{p}_u - \mathbf{p}_v\| \leqslant R$; $\|\mathbf{p}\|$ is the Euclidean norm of $\mathbf{p}$. The set of neighbours of $u \in V$ is denoted by

$$V_u = \{v \in V : \|\mathbf{p}_u - \mathbf{p}_v\| \leqslant R\}. \tag{1}$$

A group Q is represented by the tuple

$$Q = \langle l, V_l \rangle, \tag{2}$$

where l is the GL of Q and $V_l \cup \{l\}$ is the set of members of Q. We stress that $u \in V$ is a member of Q notated by $u \in Q$ and the number of members in Q including GL is notated by $|Q|$. The centroid point $\mathbf{c}_Q$ of Q is $\mathbf{c}_Q = \frac{1}{|Q|}\sum_{v \in Q} \mathbf{p}_v$. The node $l \in Q$ is appointed as the GL whose location $\mathbf{p}_l$ is the closest to $\mathbf{c}_Q$, i.e.,

$$l = \arg\min_{v \in Q} \|\mathbf{p}_v - \mathbf{c}_Q\|. \tag{3}$$

We notate by $\mathcal{Q} = GFA(V)$ the invocation of GFA with input V and output a set of groups $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_{|\mathcal{Q}|}\}$ with corresponding GLs $\{l_1, l_2, \ldots, l_{|\mathcal{Q}|}\}$ and $V_{l_i} \subset V, i = 1, \ldots, |\mathcal{Q}|$. Group Q_i is formed such that all members are within the communication range of l_i.

We propose an incremental GFA based on the Adaptive Resonance Theory adopting the concept of competitive learning [41]. The GFA takes into consideration the communication range R in order to form groups. Our algorithm starts with a single group and creates new groups as needed. The algorithm updates incrementally the centroid point $\mathbf{c}_Q$ of group Q once a node $u \in V$ is a new member for this group. Otherwise, the algorithm creates a new group Q', which has as centroid point $\mathbf{c}_{Q'}$ the new point $\mathbf{p}_u$. The threshold value either for updating the centroid of an existing group or for creating a new group depends on R. At the final stage of the algorithm, the GLs of the derived groups are elected so that Eq. (3) is satisfied. In Section 5.5 we study the fault tolerance of the considered clustering scheme. In order to guarantee reachability between all group members and their GL we set the threshold value as $0.5R$. It can be easily shown that when the threshold value is $0.5R$ then all members of the formed group are reachable from the GL (the closest member to the centroid point of the group). In the extreme cases where all members lie on a circle of radius $0.5R$ or lie on a line segment then all members are reachable by the GL. In the 'circularization' case, the GL can be any member of the group while in the 'linearization' case, the GL is located in the midpoint of the line segment.

Algorithm 1. The incremental GFA.

Input: V
Output: $\mathcal{Q}$
Parameter: R
Begin
$Q_1 \leftarrow \langle u_1, \emptyset \rangle, \mathbf{c}_{Q_1} \leftarrow \mathbf{p}_{u_1}, \mathcal{Q} \leftarrow \{Q_1\}$
for all $u \in V$ **do**
  $\mathbf{c}_{Q^*} \leftarrow \arg\min_{Q=\langle l, V_l \rangle \in \mathcal{Q}} \|\mathbf{c}_Q - \mathbf{p}_u\|$ /* $Q^* = \langle l^*, V_{l^*} \rangle$ */
  if $(\min_{Q \in \mathcal{Q}} \|\mathbf{c}_Q - \mathbf{p}_u\| \leqslant 0.5R)$ **then**
    $\mathbf{c}_{Q^*} \leftarrow \mathbf{c}_{Q^*} + 0.5(\mathbf{c}_{Q^*} - \mathbf{p}_u)$ /*update the centroid*/
    $V_{l^*} \leftarrow V_{l^*} \cup \{u\}$
  else
    $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{\langle u, \emptyset \rangle\}$ /*new group*/
  end if
end for
End

Our algorithm does not assume a predefined number of members per group. The final number of the derived groups depends on R. The incremental GFA for deriving $\mathcal{Q}$ from set V is shown in Algorithm 1. The update factor of the centroid point $\mathbf{c}_{Q^*}$ with the position $\mathbf{p}_u$ of a new member is set to 0.5 in order to have all members in the communication range of the GL.

3.2. Group cohesion metric

The cohesion metric for a group quantifies the coherency and compactness of a group and allows us to infer its persistence. The compactness of Q as defined in [5] refers to the average distance between the members' positions $\mathbf{p}_u$ and the (virtual) centroid $\mathbf{c}_Q$. We cannot directly adopt this measure in our case because:

- the centroid $\mathbf{c}_Q$ does not necessarily coincides with the real GL' position $\mathbf{p}_l$,
- the maximum distance between a member's position and GL is bounded above by R,
- the identities of the members should be the same as initially specified by the GFA.

We define as cohesion $\omega \in [0,1]$ of Q the time-varying metric:

$$\omega = \begin{cases} 0, \text{if} \exists u \in Q : \|\mathbf{p}_l - \mathbf{p}_u\| > R \\ 1 - \frac{1}{R}\left(\frac{1}{|Q|-1}\sum_{u \in V_l}\|\mathbf{p}_l - \mathbf{p}_u\|\right), \text{otherwise} \end{cases} \quad (4)$$

Actually, for a group Q_i, ω shows if the Q_i's nodes are gathered around the GL (inside the communication range R) as initially defined. The cohesion ω indicates that:

- Q consists of the same members V_l as initially identified through the GFA, i.e., it demonstrates coherency, and
- all members in Q are within the communication range of GL, i.e., it shows compactness.

A value of $\omega = 0$ indicates an invalid group while a value of $\omega \in (0,1]$ indicates a valid/coherent group. We obtain $\omega = 0$ when there is at least a member $u \in Q$ which is unreachable by the GL and $\omega > 0$ when the average distance of all members from the GL is less than (or equal to) R. The higher the ω gets, the more compact the group becomes.

Note that, the proposed cohesion metric conveys a physical meaning. Contrary to the measure in [5], when $0 < \omega \leqslant 1$, Q is both compact and coherent, i.e., all 'indigenous' members are within the communication range of the GL, thus, the GL can disseminate information to the members. The case $\omega = 0$ denotes that at least one member does not receive information from the GL, which is not addressed by the proposed measure in [5].

4. Rationale and problem formulation

In this Section, we analytically describe the scenario under consideration, we discuss our rationale and give mathematical formulations for finding the optimal stopping time. In Table 1, we provide all the notations adopted in this paper.

4.1. Rationale

Let us consider a discrete time domain $\mathbb{T} = \{0, 1, \ldots\}$. At group formation time $t_f = 0$ the GFA is invoked with V nodes and the set $\mathcal{Q}$ is derived. Let $t_{now} \in \mathbb{T}$ be the current absolute time and a group $Q \in \mathcal{Q}$ be formed at time $t_f \in \mathbb{T}$ with $t_f < t_{now}$. The GVS attempts to conclude on the persistence of Q during the period $[t_f, t_f + t_*]$ for some unknown finite horizon $t_* > 0$. Below, we use the time index t defined as $t = t_{now} - t_f$ and $1 \leqslant t \leqslant t_*$. In addition, we refer to $t_p = t_* + t_f$ as the persistence conclusion time for Q, i.e., the time instance at which Q is validated (see Fig. 4(a)).

During horizon t_*, the GVS, which is responsible for group Q (notated by $GVS(Q)$), determines whether Q retains its initial structure. The GVS at time $t \in [1, t_*]$ receives the position of each $u \in Q$ with the purpose of calculating ω_t. If the GVS is confident that Q is valid after horizon t_*, then it considers only the GL for communication and (consequently) exchanges no information with each $u \in V_l$. The system communicates with the GL in order to deliver information to the entire group (including the GL) for time instances greater than t_p. Hence, there is no need for each member to directly communicate with the system after t_p. However, the cohesion of Q is periodically checked every N time instances (checkpoints) after t_p. Hence, the system will be capable of checking the group persistence. An future extension of our work is to include a sophisticated mechanism that will adapt the value of N. For instance, we could define a feedback model that alters checkpoints based on the mobility behaviour of nodes. N affects the communication cost of the group, since all the nodes should report their position to the system. N could be tuned according to the network or computation load seen at the system or the mobility behaviour of the group.

At a checkpoint $t_c = t_p + mN, m \geqslant 1$, we distinguish two cases:

- If $\omega_{t_c} = 0$ then Q is considered fragmented. At this time, the system terminates the periodical check for this group. In the sequel, the GFA is invoked only for the set of nodes from the fragmented group Q, i.e., $\mathcal{Q}' = GFA(\{l, V_l\})$, in order to derive new clusters. Then for all $Q' \in \mathcal{Q}'$ a $GVS(Q')$ is activated.
- Otherwise, i.e., $\omega_{t_c} > 0$, Q is still valid. However, in this case, a new GL is elected for the group (different from the previous GL) based on Eq. (3). This is doable since, at each t_c all group members report their positions to the system, thus, the new centroid of the group can be re-estimated. The GL handles all communication needs when Q is at the CF state. The election of a new GL is enforced periodically (i.e., at each checkpoint as described later) in order to avoid assigning the same GL for the entire life of the group, thus, imposing a significant resource use in balance within the group.

Table 1
Notation and description.

Notation	Description
GL	Group leader
Q	Group
$\mathcal{Q}$	Set of formed groups
$u \in Q$	Node (moving object) of group Q
V	Set of nodes $\{u\}$
$\mathbf{p}_u$	Position of node u
$\mathbf{c}_Q$	Centroid of positions of nodes $u \in Q$
$\mathbf{c}_{Q,l}$	Weighted centroid of positions of nodes $u \in Q$ and node $l \in Q$
ω	Cohesion metric
R	Communication range
N	Checkpoint period
θ	Re-clustering threshold
β	Discount factor
ρ	Group density
λ_n	Cumulative number of fragmented groups up to n-th checkpoint
t_f	Formation time
t_p	Persistence conclusion time
t_*	Optimal stopping time; duration of the OS state
t_c	Checkpoint time
t_l	Last checkpoint time before group fragmentation
t_d	Fragmentation time
$t_\#$	Duration of CF state

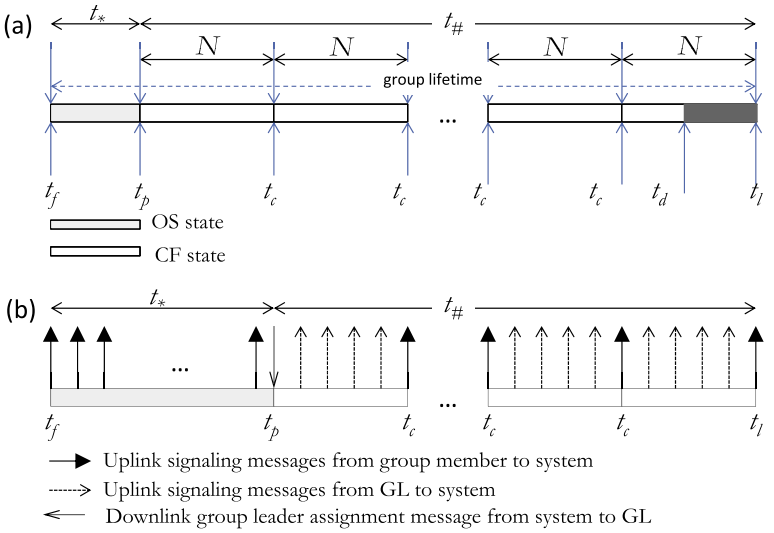


Fig. 4. (a) The group lifetime; t_f is the group formation time instance, t_p is the persistence conclusion time instance, t_c is checkpoint, t_l is the last checkpoint at which the group is detected to be fragmented, t_d is the actual fragmentation time, t_* is the duration of the OS state, $t_\#$ is the duration of the CF state, and (b) signalling between system and group members.

The 'lifetime' of a group is shown in Fig. 4(a). We notate with t_d the time instance at which Q is actually fragmented, i.e., $\omega_{t_d} = 0$. The t_l denotes the last checkpoint t_c at which the system detects that Q is fragmented, i.e., $t_d \leqslant t_l$. We denote with $t_\# = t_l - t_p$ the lifetime of a group that is considered persisting by the system.

4.2. Signalling between system and group members

The group related signalling during the group lifetime has as follows. In OS state (interval $[t_f, t_p]$; duration t_*), the system receives location updates from all members (uplink signalling communication). At t_p the system sends a GL-assignment signalling message to the GL (downlink signalling). Consequently, the GL notifies group members to stop transmitting location updates to the back-end system. In the CF state (interval $(t_p, t_l]$; duration $t_\#$), group members report their location to the system only at checkpoints t_c. The GL, while in the CF state, reports its location to the system during $[t_p + 1, t_l]$ until the group turns fragmented. Fig. 4(b) shows the discussed signalling.

When the average number of members per group falls below a certain threshold, due to the fragmentation of groups, the system (i) invokes the GFA with all the active nodes of the network, i.e., global re-clustering and creates groups anew, (ii) new GLs are elected and (iii) the validation phase for all new groups starts again. This global re-clustering is performed to avoid the continuous fragmentation of groups down to the singleton level.

4.3. Problem formulation

Consider a group Q formed at t_f, which enters the CF state. Let the GVS assert that Q is valid right after the formation time, i.e., $t_p = t_f$, thus, immediately concluding on the persistence of Q. There might be intervals in which Q is, by accident, treated as a persisting group. The GVS is fully unaware of such cases since, after the immediate election of GL, group members do not report their distinct positions to the system.

As an alternative, the GVS could wait for unknown finite horizon $t_* = t_p - t_f$ in order to be more confident on the persistence of Q. During t_*, the members of Q report their position to system, thus, the GVS can calculate the cohesion of Q. We study a GVS that postpones the validation decision through additional calculations of cohesion values. If most of cohesion values are positive and relatively high, there is a high belief that Q will be a persisting group. The GVS delays the decision up to some t_p. At t_* a decision is taken. The problem is to find the optimal stopping time t_* in order to ensure that Q is a persisting group from time instances greater than t_p.

Let Ω_t be non-negative random variables with finite mean $\mathbb{E}\Omega_t < \infty$, $t = 1, \ldots$. The $\omega_t \in [0, 1]$ is a realization value for Ω_t. We define the reward Y_t of a validation decision based on the persistence behaviour of the group. We introduce Y_t in order to treat the problem through OST. Y_t could be directly connected to the performance improvements that a timely validation decision yields. Y_t is a random variable generated by the sum of cohesion values up to $t, S_t = \sum_{k=1}^{t} \Omega_k$, discounted by a factor $\beta \in (0, 1)$, i.e.,

$$Y_t = \beta^t S_t. \tag{5}$$

The GVS has to find t_* in order to stop calculating the cohesion of Q and after that consider Q as a persisting group. If the GVS never stops, then the reward is zero, $Y_\infty = 0$. Furthermore, the GVS will never stop at t with $S_t \leqslant 0$. As the cohesion assumes high values for certain time instances, then S_t increases at a high rate, thus, indicating a compact group. The problem is to decide how large the S_t should get before the GVS stops. Hence, the problem is to find a time instance t such that the supremum in Eq. (6) is attained.

$$\sup_t \mathbb{E}(Y_t). \quad (6)$$

A high β value indicates a conservative GVS. This means that the GVS requires a rather extended time interval for concluding on the persistence of Q. This, however, comes at the expense of high communication load since, during this period, all members undertake regular position reporting. A low β value denotes a rather optimistic GVS. It is likely that an optimistic GVS reaches premature decisions on group persistence. The problem in Eq. (6) is treated as an infinite horizon OST problem with discounted future reward [42].

5. Optimally scheduled group validation system

In this section, we propose the Optimally Scheduled GVS (OGV) that solves the problem in Eq. (6). Initially, we briefly discuss OST. Then, we prove the existence of t_*, report on the corresponding optimal stopping rule and elaborate on the OGV scheme.

5.1. Optimal stopping theory

The OST deals with the problem of choosing the best time instance to take the decision of performing a certain action. This decision is based on sequentially observed random variables in order to maximize the expected reward [12]. For given random variables $X_1, X_2, \ldots$ and measurable functions $Y_t = \mu_t(X_1, X_2, \ldots, X_t)$, $t = 1, 2, \ldots$ and $Y_\infty = \mu_\infty(X_1, X_2, \ldots)$, the problem is to find a stopping time τ to maximize $\mathbb{E}Y_\tau$. The τ is a random variable with values in $\{1, 2, \ldots\}$ such that the event $\{\tau = t\}$ is in the Borel field (filtration) $\mathbb{F}_t$ generated by $X_1, \ldots, X_t$, i.e.,

$$\mathbb{F}_t = \mathbb{B}(X_1, \ldots, X_t) \quad (7)$$

Hence, the decision to stop at t is a function of $X_1, \ldots, X_t$ and does not depend on future observables $X_{t+1}, \ldots$. The theorem in [43] refers to the existence of the optimal stopping time.

Theorem 1 [43]. *If $\mathbb{E}(\sup_t Y_t) < \infty$ and $\lim_{t\to\infty} \sup Y_t \leqslant Y_\infty$ almost surely (abbreviated as a.s.) then the stopping time $t_* = \inf\{t \geqslant 1 | Y_t = ess\sup_{\tau \geqslant t} \mathbb{E}(Y_\tau | \mathbb{F}_t)\}$ is optimal.*

Proof. See [43]. □

The (essential) supremum $ess\sup_{\tau \geqslant t} \mathbb{E}(Y_\tau | \mathbb{F}_t)$ is taken over all stopping times τ such that $\tau \geqslant t$ a.s. The optimal stopping time t_* is obtained through the 'principle of optimality' [47].

5.2. An optimal stopping time for group validation

The OGV is applied for every group Q. OGV concludes on the persistence of Q based on sequential calculations (observations) of ω_t values. Persisting high values of $\omega_1, \omega_2, \ldots, \omega_t$ makes the OGV confident on the persistence of Q. On the other hand, as the OGV delays its decision, the validation process progresses further, computational effort and communication load increase. A decision taken at time instance t is:

- either to assert Q as a persisting group and, then, stop the validation process (Q transits from the OS to the CF state); decision D1,
- or continue the validation process at time $t+1$ with additional communication load (Q remains at the OS state); decision D2.

We will show that the OGV bases its decision at t on $S_t = \sum_{k=1}^{t} \Omega_k$.

Theorem 2. *An optimal stopping time for the problem in Eq. (6) exists.*

Proof. Based on the theorem in [43], we have to prove that the optimal stopping time t_* exists and is derived from the principle of optimality. Specifically, we have to prove that (i) $\lim_{t\to\infty} \sup Y_t \leqslant Y_\infty$ a.s. and (ii) $\mathbb{E}(\sup Y_t) < \infty$.

Note that Ω_t are non-negative and from the strong law of numbers $(\frac{1}{t})\sum_{k=1}^{t} \Omega_k \to \mathbb{E}\Omega$ a.s., so that

$$Y_t = t\beta^t (S_t/t) \leqslant t\beta^t (1/t) \sum_{k=1}^{t} \Omega_k \simeq t\beta^t \mathbb{E}\Omega \xrightarrow{a.s.} 0$$

with $\lim_{t\to\infty} \sup Y_t = Y_\infty = 0$.

In addition, $\sup_t Y_t = \sup_t \beta^t S_t \leqslant \sup_t \beta^t \sum_{k=1}^{t} \Omega_k \leqslant \sup_t \sum_{k=1}^{t} \beta^k \Omega_k \leqslant \sum_{k=1}^{\infty} \beta^k \Omega_k$. Hence,

$$\mathbb{E}(\sup_t Y_t) \leqslant \sum_{k=1}^{\infty} \beta^k \mathbb{E}\Omega = \mathbb{E}\Omega \frac{\beta}{1-\beta} < \infty. \qquad \square$$

It is worth noting that our problem as expressed in Eq. (6) is equivalent to the 'burglar problem' discussed in [48]. A burglar contemplates a series of burglaries. She may accumulate her earnings as long as she is not caught, but if caught during a burglary, she loses everything including her initial fortune, if any, and she is forced to retire. She wants to retire before being caught. In this problem, β represents the probability that a burglar is not getting caught at the t-th burglary [48] notated by the event $\{Z_t = 1\}$ with $P(Z_t = 1) = \beta$. In addition, the returns (Ω_t in our case) for each burglary are independent of the 'arrest' event $\{Z_t = 0\}$, which is on each trial equally probable with probability $P(Z_t = 0) = 1 - \beta$. It is assumed that if the burglar is ever caught, her return is fixed at zero. The burglar's problem is to choose a stopping time τ to maximize her expected fortune $\mathbb{E}(\prod_{k=1}^{\tau} Z_k \sum_{k=1}^{\tau} \Omega_k)$.

In our case, Ω_t are non-negative, thus, the problem is *monotone* [49]. In such case the optimal stopping time t_* is obtained by the *1-stage look-ahead optimal rule* (1-sla) [50]: stop at the *first* stopping time t at which $Y_t \geqslant \mathbb{E}(Y_{t+1} | \mathbb{F}_t)$, i.e.,

$$t_* = \inf\{t \geqslant 1 | Y_t \geqslant \mathbb{E}(Y_{t+1} | \mathbb{F}_t)\}$$

In our context, for a monotone stopping problem with observations $\Omega_1, \Omega_2, \ldots$ and rewards $Y_1, Y_2, \ldots, Y_\infty$, the 1-sla is optimal since $\sup_t Y_t$ has finite expectation $(\mathbb{E}\Omega \frac{\beta}{1-\beta})$ and $\lim_{t\to\infty} \sup Y_t = Y_\infty = 0$ a.s. (see Theorem 2). The interested reader could also refer to Appendix A for the characteristics of Y_t.

Theorem 3. *The optimal stopping time t_* for the problem in Eq. (6) is*

$$t_* = \inf\left\{t \geqslant 1 | S_t \geqslant \frac{\beta}{1-\beta}\mathbb{E}\Omega\right\}. \quad (8)$$

Proof. Let $Z_1, Z_2, \ldots$ be i.i.d. random variables with $P(Z_t = 1) = \beta$ and $P(Z_t = 0) = 1 - \beta$. The reward for D1, i.e., stopping after the t-th realization of Ω_t with ω_t, is

$$Y_t = \left(\prod_{k=1}^{t} Z_k\right)\sum_{k=1}^{t}\Omega_k \quad (9)$$

for $t = 1, 2, \ldots$ Consider the filtration $\mathbb{F}_t$ generated by both $\Omega_1, \Omega_2, \ldots, \Omega_t$ and $Z_1, Z_2, \ldots, Z_t$. Based on the principle of optimality [47] and the 1-sla, OGV should stop the validation process at t if

$$Y_t \geqslant \mathbb{E}(Y_{t+1}|\mathbb{F}_t) \quad (10)$$

We evaluate t_* through calculating the $\mathbb{E}(Y_{t+1}|\mathbb{F}_t)$ on $\prod_{k=1}^{t} Z_k = 1$. We obtain that

$$\mathbb{E}(Y_{t+1}|\mathbb{F}_t) = \mathbb{E}\left(Z_{t+1}\sum_{k=1}^{t+1}\Omega_k|\mathbb{F}_t\right) = \mathbb{E}(Z_{t+1}S_{t+1}|\mathbb{F}_t) \quad (11)$$

Since $S_t = \sum_{k=1}^{t}\Omega_k$ then $Y_t = S_t$ and $S_{t+1} = S_t + \Omega_{t+1}$, we obtain from Eq. (11) that:

$$\begin{aligned}\mathbb{E}(Y_{t+1}|\mathbb{F}_t) &= \mathbb{E}(Z_{t+1}|\mathbb{F}_t)(\mathbb{E}(S_t|\mathbb{F}_t) + \mathbb{E}(\Omega_{t+1}|\mathbb{F}_t)) \\ &= \mathbb{E}(Z_{t+1})(S_t + \mathbb{E}(\Omega_{t+1})) = \beta(S_t + \mathbb{E}\Omega)\end{aligned}$$

since $S_t \in \mathbb{F}_t$ and Ω_{t+1} is independent of the filtration $\mathbb{F}_t$. Hence, $\mathbb{E}(Y_{t+1}|\mathbb{F}_t) = \beta(S_t + \mathbb{E}\Omega)$. The optimal stopping time is then $t_* = \inf\{t \geqslant 1 | Y_t \geqslant \mathbb{E}(Y_{t+1}|\mathbb{F}_t)\}$, i.e., when $Y_t = S_t \geqslant \mathbb{E}(Y_{t+1}|\mathbb{F}_t) = \beta(S_t + \mathbb{E}\Omega)$, or

$$t_* = \inf\left\{t \geqslant 1 | S_t \geqslant \frac{\beta}{1-\beta}\mathbb{E}\Omega\right\} \quad \square$$

Note that we can incrementally calculate $\mathbb{E}\Omega$, e.g., through incremental kernel density estimation from sequence $\omega_1, \omega_2, \ldots, \omega_t$ [45,46]. In Appendix B, we provide an incremental method for $\mathbb{E}\Omega$ calculation.

Eq. (8) states that the validation process stops at the first t (after the group formation time) when S_t is at least $\frac{\beta}{1-\beta}\mathbb{E}\Omega$. The OGV stops calculating the cohesion for Q once the criterion in Eq. (8) holds true and, then, Q enters the CF state. Algorithm 2 refers to the proposed $OGV(Q)$ algorithm for the OS and CF states of Q. The OGV algorithm is terminated when Q is fragmented.

Algorithm 2. The OGV algorithm.

Input: Q
Parameters: β, N

Begin
$t \leftarrow 1, stopped \leftarrow FALSE, S_t \leftarrow 0$
/*The OS state*/
while ($\neg$ *stopped*) **do**
  calculate ω_t using Eq. (4)
  update $\mathbb{E}\Omega$
  $S_t \leftarrow S_t + \omega_t$
  if ($S_t > \frac{\beta}{1-\beta}\mathbb{E}\Omega$) **then** $stopped \leftarrow TRUE, t_* \leftarrow t$
  else $t \leftarrow t + 1$ **end if**
end while
/*The CF state*/
$fragmented \leftarrow FALSE$
while ($\neg$ *fragmented*) **do**
  if (t mod $N \equiv 0$) **then**/*checkpoints*/
    if ($\omega_t \equiv 0$) **then** $fragmented \leftarrow TRUE$
    end if
  end if
  $t \leftarrow t + 1$
end while
End

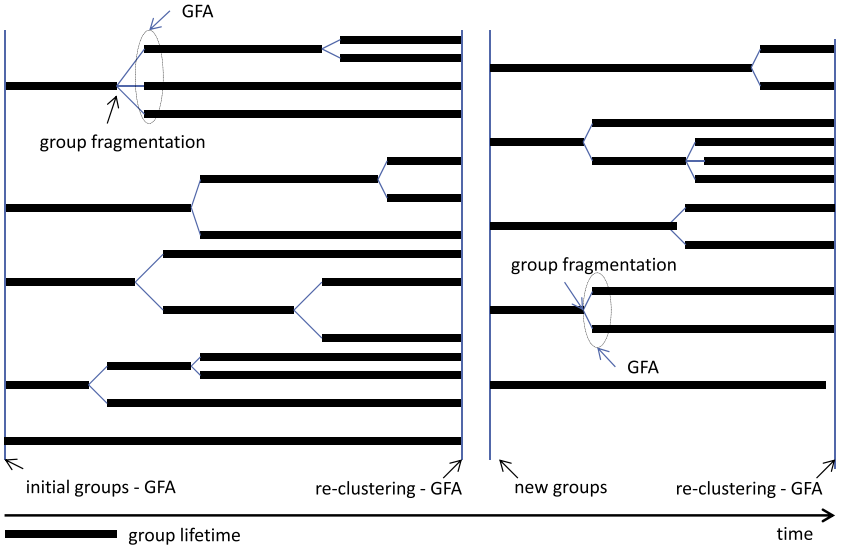


Fig. 5. The lifetime of groups. Each time a group gets fragmented the GFA is invoked. Once $\rho > \theta\rho_0$ all nodes are re-clustered (global re-clustering event) through the GFA.

It is worth noting that in our case ω_t correlates with the ω_{t-1} due to the movement of (group) members. In order to exploit such correlation between ω_t and ω_{t+1}, the problem in Eq. (6) would be of the form $J_t(Y_t) = \max\{\beta^t \sum_{k=1}^{t} \Omega_k, \mathbb{E}[J_{t+1}(H_{t+1}(Y_t, O_t))]|\mathbb{F}_t\}$. $J_t(\cdot)$ is the reward function at t and O_t are i.i.d. random variables taking positive values with given probability distribution. Hence, Y_t requires knowledge on the $H_{t+1}(\cdot,\cdot)$ for each t with analytical formulae for each correlation function H_{t+1}. In [47] (Ch. 4) and [51] it is shown that in the case of correlated observations the optimal stopping rule is also of the same type as the one for i.i.d. cohesion values (i.e., a criterion-based solution). In order to obtain an optimal stopping criterion, one has to find an optimal stopping rule as the solution of the equation: $Y_t = \mathbb{E}\{J_{t+1}(H_{t+1}(Y_t, O_t))\}, t \geqslant 1$. In this work, for simplicity, we assume i.i.d. cohesion values. For completeness, we provide an optimal criterion assuming linear correlation between ω_t and ω_{t-1} in Appendix C. We could proceed with modelling the dynamics of our system as a non-linear stochastic dynamic system. This implies that one has to learn first the correlation function of the system dynamics, e.g., through adopting kernel predictive linear Gaussian models [64] or multiple-order kernel autoregression model. However, this poses significant limitations. Firstly, the complexity of learning such dynamics is prohibitive for fast, on-line detection of the coherency of groups. Secondly, the derived stochastic equation from the trained model, although computationally intractable, does not guarantee the application of the 1-sla optimal rule and, thus, the existence of an optimal stopping time. The reason is that we should handle multiple computations of convolutions of the probability density function of $\omega, P(\omega)$ (also unknown). In that case, the optimal stopping time does not exist. Finally, the property of classifying an optimal stopping time problem as a 'monotone' problem derived from a (non-linear) stochastic equation is not an easy task (and beyond of the scope of this paper). To this end, our decision is to model our problem as a monotone, 1-sla problem, where we prove the existence of an optimal stopping time by adopting linearity between successive observations.

5.3. Group fragmentation incidents and global re-clustering

We discuss the application of the OGV to all groups in $\mathcal{Q}$. Let t_f be the formation time of the groups in $\mathcal{Q}$. For each Q an instance of $OGV(Q)$ is responsible for the validation of Q. Each time a Q becomes fragmented its members are clustered into new groups $\mathcal{Q}'$ through the invocation of the GFA, i.e., $\mathcal{Q}' = GFA(\{l, V_l\})$. Hence, the current set of groups increases by the new groups excluding the fragmented Q, i.e., $\mathcal{Q} \leftarrow (\mathcal{Q} \setminus \{Q\}) \cup \mathcal{Q}'$. In the sequel, instances of $OGV(Q')$ are considered responsible for each $Q' \in \mathcal{Q}'$. The fragmentation of a group leads (i) to a decrease of the mean number of members per group, hereinafter, referred to as *group density*:

$$\rho = \frac{1}{|\mathcal{Q}|} \sum_{Q \in \mathcal{Q}} |Q| \tag{12}$$

and (ii) to an increase in the total number of groups. In order to avoid the case of groups with only one member (the GL), a fragmented group forms (sub) groups iff $\rho \leqslant \theta\rho_0$, where ρ_0 is the initial group density right after a global re-clustering, i.e., GFA invocation over V for some re-clustering threshold $\theta \in (0, 1)$. The higher θ gets, the more frequent a re-clustering event becomes. If ρ is greater than a fraction of ρ_0, the GFA is invoked only for the

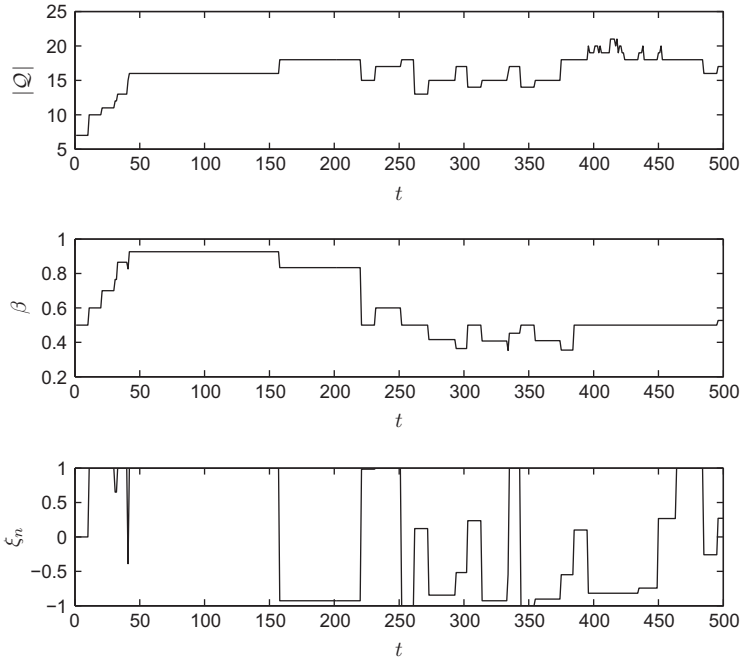


Fig. 6. The $|\mathcal{Q}|, \beta$ and ξ_n values vs. time from the adaptation mechanism.

fragmented group(s). Otherwise all nodes are subject to global re-clustering through the invocation of $GFA(V)$ and a new set $\mathcal{Q}$ is created. After a re-clustering event, the OGV is initiated for each newly derived $Q \in \mathcal{Q}$.

It is worth noting that at a re-clustering event the OGV for each group, which was not fragmented, quits the validation process. Algorithm 3 is the algorithm of the LBS group-oriented system which deals with the application of OGV to all groups in $\mathcal{Q}$. Fig. 5 shows the lifetime of the groups.

5.4. The adaptation mechanism

The β factor is not known a priori for OGV invocations. The β has to be adjusted w.r.t. the fragmentation behaviour of groups. If a certain number of groups turn fragmented within a short time horizon then the system could prolong the validation process. That is because the system takes pre-mature decisions on group persistence, thus, it could prolong the group validation by increasing β. However, prolongation of the validation process results to an increase of uplink and downlink signalling messages. If the number of fragmented groups is relatively small for a long time horizon, then the system shortens the validation time horizon by decreasing β. Low β results to reduced uplink and downlink signalling messages, since the group is treated as persisting early in the validation process.

Algorithm 3. The algorithm of the group-oriented LBS System

Input: V
Parameter: θ

Begin
reclustering $\leftarrow$ *TRUE*
while (true) **do**
  if (*reclustering*) **then**
    $\mathcal{Q} \leftarrow GFA(V)$
    reclustering $\leftarrow$ *FALSE*
    for all $Q \in \mathcal{Q}$ **do**
      new $OGV(Q)$
    end for
  end if
  $\rho_0 \leftarrow \frac{1}{|\mathcal{Q}|}\sum_{Q\in\mathcal{Q}}|Q|$
  for all $Q = \langle l, V_l\rangle \in \mathcal{Q}$ **do**
    if (*fragmented*(Q)) **then**
      $\mathcal{Q}' \leftarrow GFA(\{l, V_l\})$
      for all $P \in \mathcal{Q}'$ **do**
        new $OGV(P)$
      end for
      $\mathcal{Q} \leftarrow (\mathcal{Q} \setminus \{Q\}) \cup \mathcal{Q}'$
    end if
  end for
  $\rho \leftarrow \frac{1}{|\mathcal{Q}|}\sum_{Q\in\mathcal{Q}}|Q|$
  if$(\rho > \theta\rho_0)$ **then**
    reclustering $\leftarrow$ *TRUE*
  end if
end while
End

We propose an adaptation mechanism for β value determination based on the rate of group fragmentation. Note that the system checks if Q is fragmented only at checkpoints. Let $\mathcal{F}_{\tau_i}$ be the set of groups that are identified by the system as being fragmented at checkpoint $\tau_i \in \mathbb{T}$, $i = 1, \ldots, n$, where n is the current number of checkpoints. Consider the quantity

$$\lambda_n = \frac{1}{\tau_n - \tau_{n-1}}\left(\frac{1}{n}\sum_{i=1}^{n}|\mathcal{F}_{\tau_i}|\right), \tag{13}$$

where $\frac{1}{n}\sum_{i=1}^{n}|\mathcal{F}_{\tau_i}|$ is the average number of fragmented groups up to τ_n. The variable λ_n quantifies the cumulative number of fragmented groups up to n-th checkpoint with a weight reciprocal to the duration $\tau_n - \tau_{n-1}$ of the last two checkpoints, i.e., $(n-1)$-th and n-th checkpoint. A high λ_n value indicates that at the time interval $[\tau_{n-1}, \tau_n]$ a significant number of groups have turned fragmented with high rate provided that $\tau_n - \tau_{n-1}$ is relatively small. Let ξ_n denote the relative increase/decrease of the number of fragmented groups, i.e., $\xi_n = \frac{\lambda_n - \lambda_{n-1}}{\lambda_{n-1}}$. A high positive ξ_n value indicates an increased rate in the number of fragmented groups. In this case, one can further prolong the validation process by increasing β to avoid 'immature' decisions. Recall that an increase in β refers to the case where the system delays an optimal decision in order to be certain enough about group persistence. On the other hand, a high negative ξ_n value refers to the case where we obtain a decrease in the fragmentation rate of groups. In this case the β value decreases, thus, the system does not delay in making a decision provided that a group exhibits a persisting behaviour. The β value at the n-th checkpoint is updated through the rule:

$$\beta_n \leftarrow (1 + \text{sign}(\xi_n) \cdot \min\{|\xi_n|, 1\}) \cdot \beta_{n-1}. \tag{14}$$

The β in Eq. (14) increases/decreases linearly by ξ_n. If $\beta_n \leqslant 0$ then $\beta_n = \epsilon$, and, if $\beta_n \geqslant 1$ then $\beta = 1 - \epsilon$; ϵ is a relatively small positive number (e.g., $\epsilon = 0.01$).

Upon invocation of $OGV(Q)$, β gets updated using Eq. (14) and remains constant until validation of Q. Initially (at t_f) $\beta = 0.5$ and, at every re-clustering event, β resets to 0.5. Fig. 6 shows the current number of groups $|\mathcal{Q}|$, the current β and ξ_n against time. We observe that β assumes rapidly high values at the early stages. At that time interval $|\mathcal{Q}|$ increases with high rate.

5.5. System robustness

In order to safeguard the uninterruptible operation of the proposed scheme, despite possible failures of the appointed GLs (e.g., run out of energy), we introduce two strategies: the 'soft' and 'hard' GL substitution.

5.5.1. 'Soft' GL substitution

In this strategy, we define the extra role of GL deputies. A subset of group members are appointed as deputies and may undertake the GL responsibilities following a GL failure event. Right after the group formation, a member $u \in Q$ is appointed as deputy if $\|\mathbf{p}_u - \mathbf{p}_l\| \leqslant \phi R$, where l is the current GL and some positive ϕ (e.g., $\phi \in (0, 0.5]$). The

Table 2
Performance metrics.

Notation	Description	Desired value
$c_Q \in \mathbb{R}$	Total communication load	Low
$\alpha_Q \in (0,1]$	Load ratio	Low
$\delta_Q \in [0,1]$	Lowest possible load guaranteeing highest information accuracy	low
$\gamma_Q \in [0,1]$	information accuracy	High
$w \in [0,1]$	Global re-clustering events	Low
$f \in [0,1]$	Percentage of members associated to fragmented groups	Low
MDP $\in [0,1]$	Mean percentage delay	N/A

GL deputies form the subset $Q_\phi \subseteq Q$. As ϕ increases, the percentage of GL deputies in the group population increases too. There is a trade-off between fault tolerance and network overhead. As ϕ approaches unity, the network load is negatively impacted assuming that uplink and downlink signalling communication involves not only the GL but also all the appointed GL deputies. As ϕ approaches zero, the fault tolerance of the scheme is undermined.

GL deputies perform exactly the same tasks as their GL, i.e., perform uplink signalling communication during the OS and CF states. Let $\mathbf{c}_{Q_\phi}$ be the centroid of the GL deputies set Q_ϕ, i.e.,

$$\mathbf{c}_{Q_\phi} = \frac{1}{|Q_\phi|}\sum_{u\in Q_\phi}\mathbf{p}_u \tag{15}$$

and $\mathbf{c}_{Q_\phi,l}$ be the weighted centroid of the GL deputies set Q_ϕ including the GL l, i.e.,

$$\mathbf{c}_{Q_\phi,l} = \frac{|Q_\phi|\mathbf{c}_{Q_\phi} + \mathbf{p}_l}{|Q_\phi|+1}. \tag{16}$$

In the event of GL l failure, the GL deputy $l^* \in Q_\phi$, whose $\mathbf{p}_{l^*}$ is the closest point to $\mathbf{c}_{Q_\phi,l}$, i.e.,

$$l^* = \arg\min_{u\in Q_\phi}\|\mathbf{c}_{Q_\phi,l} - \mathbf{p}_u\|,$$

is appointed as the new GL of the group. Intuitively, such an option yields a lower probability of having some group members disconnected (unreachable by the new GL), since the new GL at the time of the GL failure is located both closest to the (previous) GL and the rest GL deputies.

In order to assess the robustness of this strategy, we define the probability P_ϕ that at least one deputy from Q_ϕ is alive. We assume that a group member is alive with some probability $p > 0$. Hence, $P_\phi = 1-(1-p)^{|Q|_\phi}$ gives the probability at least one deputy from Q_ϕ is a GL of Q; $|Q|_\phi$ counts the deputies of Q. In Appendix D, we calculate P_ϕ for a set of traces and certain ϕ and p values.

5.5.2. 'Hard' GL substitution

We should stress that the altruistic assumption of a GL appointment does not imply that the GL is always capable to be (fully) operational, e.g., the infinite energy assumption does not hold true. Specifically, in the event of the GL failure, for instance GL runs out of energy, GL can either notify the system prior to shutdown (graceful removal/shutdown) or not. In the former case, once GL identifies that e.g., it runs out of energy, it notifies the system for no longer being operational. In the latter case the system (at the time instance following the GL shutdown) identifies that GL is not active since it does not receive any location update by the GL. In both cases, the system forces group members to report their position, thus, being able to determine the new GL. In the case where the group is identified as fragmented the system acts as discussed in Section 5.3.

6. Performance and comparative assessment

6.1. Performance metrics

We define the performance metrics in order to assess and compare the behaviour of our scheme in terms of communication load, gain, and information accuracy. Consider the lifetime of a group Q in Fig. 4(a). The number of messages exchanged during the lifetime of Q are:

- $|Q|\cdot t_*$ during the OS state. It refers to location reports from nodes to the system.
- $\frac{|Q|}{N}\cdot t_\#$ during the CF state. It refers to the periodical checks with checkpoint period N performed by the system receiving the location reports from all group members.
- $1\cdot t_\#$ during the CF state. It refers to the GL's location report to the system.
- a GL-assignment message. It refers to the downlink signalling message from system to a node, which the latter is appointed as the GL of the group.
- $|Q|-1$ from the GL to group members. It refers to the intra-group signalling from GL to each node in order for the latter to stop communicating with the system after the GL assignment.

The total communication load c_Q is then

$$c_Q = |Q|(1+t_*) + \left(\frac{|Q|}{N}+1\right)t_\#. \tag{17}$$

The communication load for Q would be $|Q|\cdot(t_*+t_\#)$ if the $OGV(Q)$ was not activated.

We denote as $\alpha_Q \in (0,1]$

$$\alpha_Q = \frac{c_Q}{|Q|(t_*+t_\#)} \tag{18}$$

the ratio of c_Q out of the communication load if the $OGV(Q)$ is not activated. A low α value indicates that we achieve low communication load.

Consider the set of the fragmented groups $\mathcal{F}$. Given a $Q\in\mathcal{F}$ there are members $v\in V_l'\subseteq V_l$ which are not within the communication range of the GL. The quantity $t_l - t_d$ (see Fig. 4(a)) indicates the elapsed time from the actual fragmentation time t_d to the detection time t_l (i.e., the last checkpoint). A high $(t_l - t_d)$ duration indicates that the OGV delays in detecting Q's fragmentation. During the period $[t_d, t_l]$ the system is unaware that the members in V_l' are not reachable by their GL. Provided that the OGV were not performed, the communication load that the system would induce from the members in V_l' is $(t_l - t_d)|V_l'|$. This load is

Table 3
Real and synthetic movement traces details.

Real movement traces	Details
Number of datasets	5
Number of nodes	100
Duration	1213 h
Area type	University campus (T0, T1), Manhattan city (T2), Disney world (T3), Market place (T4)
Synthetic movement traces	**Details**
Number of datasets	5
Number of nodes	500
Duration	1000
Area	500 m^2
Group mobility model	RPGM [52]
RPGM speed deviation (m/s)	$\sim U(0.2, 1.4)$
RPGM angle deviation (°)	$\sim (45, 90)$
GL mobility model (1)	RWP [54]
RWP speed (m/s)	$\sim U(2, 4)$
RWP pause time (s)	$\sim U(0, 10)$
GL mobility model (2)	LW [55]
LW flight lengths and pause time	$\sim Levy(a, b), a = 1.6, b = 0.8$
Manhattan speed (m/s)	$\sim U(5, 10)$
Manhattan pause time (s)	$\sim U(0, 10)$
Manhattan acceleration (m/s)	$\sim U(0, 2)$
Manhattan distance ratio (m)	$\sim U(0, 2)$
GL mobility model (3)	Manhattan [62]
RW speed (m/s)	$\sim U(5, 10)$
RW pause time (s)	$\sim U(0, 10)$
RW distance (m)	10
GL mobility model (4)	RW [63]
PRW speed (m/s)	$\sim U(5, 10)$
PRW pause time (s)	$\sim U(0, 10)$
PRW distance (m)	10
PRW probabilities	[0.0, 0.5] and [0.3, 0.7]
GL mobility model (5)	PRW

the lowest possible in order to guarantee that the members in $V_{l'}$ do not lose any information provided by the system. The metric

$$\delta_Q = \frac{(t_l - t_d)|V'_l|}{(t_\# + t_*)|Q|} \tag{19}$$

denotes the portion of the communication load that the system would have incurred in order to guarantee that the members in V'_l are provided with information from the system. The $\delta_Q \in [0, 1]$ and a low value of δ_Q indicates that (i) $|V'_l|$ is relatively small (w.r.t. $|Q|$) and/or (ii) the $(t_l - t_d)$ duration is relatively small (w.r.t. N). The system should assume a low value of δ_Q for all $Q \in \mathcal{F}$.

We define as gain g the product

$$g = \left(1 - \frac{1}{|\mathcal{Q}|}\sum_{Q\in\mathcal{Q}}\alpha_Q\right)\left(1 - \frac{1}{|\mathcal{F}|}\sum_{Q\in\mathcal{F}}\delta_Q\right) \tag{20}$$

where $\mathcal{Q}$ and $\mathcal{F}$ is the set of all formed and fragmented groups, respectively. Gain g indicates the return from the decrease in communication load achieved through the OGV ($\alpha = \frac{1}{|Q|}\sum_{Q\in\mathcal{Q}}\alpha_Q$) discounted by $1 - \delta_Q$ for all $Q \in \mathcal{F}$. The aim is to have a high g values that indicates that the model tastes low communication load for messages exchange between the group and the system.

During the CF phase, the members of Q should receive the information sent from the system, if any, through their GL. The information loss indicator I_t at time t $(1 \leqslant t \leqslant t_* + t_\#)$ is:

$$I_t^Q = \begin{cases} 0 & \text{if} Q \text{is in the OS state} \\ \frac{|V'_{l,t}|}{|Q|} & \text{otherwise} \end{cases}$$

where a member is not reachable by the GL at time t. $I_t^Q = 0$ if Q is in the OS state, thus, no member loses information from the system. Otherwise (i.e., Q is in the CF state), $I_t^Q \in (0, 1]$, which denotes the portion of members of Q that are not reachable by the GL. Such members lose information coming from the system. We define information accuracy $\gamma_Q \in [0, 1]$ during the lifetime of Q as:

$$\gamma_Q = \frac{1}{t_* + t_\#}\sum_{1\leqslant t\leqslant t_*+t_\#}(1 - I_t^Q). \tag{21}$$

High γ_Q denotes that most of the members of Q do not lose information coming from the system.

We also measure the number of (global) re-clustering events $w \in [0, 1]$ within a given time horizon T indicating the frequency of a re-clustering event, which is performed to avoid the continuous fragmentation of groups down to the singleton level. In addition, we measure the portion f of the members of the fragmented groups in $\mathcal{F}$ out of the total number of nodes $|V|$, i.e.,

$$f = \frac{1}{|V||\mathcal{F}|}\sum_{Q\in\mathcal{F}}|Q|. \tag{22}$$

Finally, we measure the *mean percentage delay* (MPD), i.e.,

$$MPD = \frac{t_*}{t_* + t_\#}, \quad (23)$$

of groups being in the OS state. A high MPD value indicates that the OGV delays the validation process in order to avoid pre-mature decisions for group persistence. Table 2 summarizes the performance metrics along with the corresponding desired value.

6.2. Simulation setup

In this section, we report on the traces and parameters we used to evaluate the performance of the OGV. We experimented with real and synthetic traces.

6.2.1. Real movement traces

The real traces refer to human walks observed within a radius of tens of kilometers. The CRAWDAD data sets[2] correspond to 91 experiment participants carrying GPS receivers. The trace collection lasted for September 2006 to January 2007. The spatial scope of the experiment includes two university campuses (one in Asia -the Korea Advanced Institute of Science and Technology (KAIST) university campus; trace T0, and one in the US -the North Carolina State University (NCSU) campus; trace T1), one metropolitan area (New York city; trace T2), one theme park (Disney World, Florida, USA; trace T3), and one State fair (North Carolina state fair; trace T4). The participants walk most of the times in these locations and may also, occasionally, travel by bus, cars, or subway trains. The total duration of traces taken over the five different sites are over 1000 h. The GPS receivers report the current position every 10 s.

The T0 traces are collected by 32 students who live in a KAIST campus dormitory. The T1 traces are taken by 20 students who took a course in the Computer Science department at NCSU. The mobility traces refer to their daily regular activities. The T2 traces were obtained from 21 persons living in Manhattan. The participants have offices in Manhattan and their means of travel include subway trains, buses, and mostly walking. The T3 traces were obtained from 19 persons who spent their holidays in Disney World, Florida, USA. The traces refer to track logs from the inside of the theme parks. The participants mainly walked in the parks. The T4 traces were obtained from 10 persons who visited a local state fair that includes many street arcades, small street food stands and showcases. The site is completely outdoor.

6.2.2. Synthetic movement traces

The synthetic traces generated by the Reference Point Group Mobility model (RPGM) [52]. In RPGM each group has a logical centre, i.e., the GL, whose motion defines the behaviour and the trajectory of the whole group. The RPGM traces were generated by the MobiSim tool [53]. MobiSim defines a mobility model for the movement of the GLs and outputs a mobility trace for all members in all groups. We generate traces for groups in which the mobility model for the GL is (i) the Random Way Point mobility model (RWP) [54], the (ii) the Levy Walk mobility model (LW) [55], (iii) the Manhattan mobility model [62], (iv) the Random Walk (RW) mobility model [63] and (v) the Probabilistic RW (PRW) mobility model.

In RWP the speed of the GL ranges in the interval $[2, 4]$ m/s and the pause ranges in the interval $[0, 10]$ s. The authors in [56] study the outdoor human mobility patterns and conclude that the mobility patterns of the participants in these outdoor settings demonstrate behaviour compatible with the *Levy walks*. A *flight* in a Levy walk is defined to be a longest straight line trip from one location to another that a 'particle' makes without a directional change or pause. Levy walks consist of many short flights and exceptionally long flights. Moreover, the flight distributions and pause time distributions closely match truncated power-law distributions. We adopt the LW for the GL in which a step is represented by: the flight length, direction, flight time, and pause time. The LW selects flight lengths and pause times randomly from Levy distributions with coefficients $a = 1.6$ and $b = 0.8$, respectively, as adopted in [56]. The direction of the GL is randomly selected from a uniform distribution of angle within [0, 360] degrees. In RPGM the speed deviation value and the angle deviation value for all members in a group ranged in the interval $[0.2, 1.4]$ m/s and $[45, 90]$ degrees, respectively [57]. Unlike RWP mobility, Manhattan mobility model uses a grid road topology. Initially, nodes are placed randomly of the edge of the graph. Accordingly, they are moved towards to randomly chosen destinations employing a probabilistic approach in the selection of nodes movements. For simulations adopting the Manhattan mobility model, we create traces with maximum acceleration in the interval $(0, 2]$, the speed in the nodes in the interval $[5, 10]$ and pause time in $(0, 10]$. The RW mobility model was first presented by Einstein in 1926. The model tries to mimic the unpredictable movements of objects in nature. In RW, a node moves from its current location to a new location by randomly choosing a direction and speed in which to travel. Each movement occurs in either a constant time interval or a constant distance travelled at the end of which a new direction and speed are calculated. Parameters of the RW traces are as follows: pause time in $[0, 10]$, GL speed in $[5, 10]$ and distance equal to 10. The PRW adopts probabilities to determine the next position of a mobile node. A probability matrix is adopted to define probabilities for moving to different directions. Once the direction is determined, the node travels with a fixed speed for a specified time interval. The parameters of the PRW are the same as in the RW model and the probabilities adopted by the PRW are in the intervals $[0.0, 0.5]$ and $[0.3, 0.7]$.

We generated 1000 synthetic traces, each with $|V| = 500$ members and each group initially consisted of $|Q| = 20$ members. The simulation time for the synthetic traces is $T = 1000$. All simulations were performed in a 500 m $\times$ 500 m plane. We refer to the synthetic trace in which the GL adopts the LW, RWP, Manhattan, RW and PRW as S0, S1, S2, S3 and S4 traces, respectively. Table 3 summarizes the details of the real and synthetic movement traces.

[2] http://crawdad.cs.dartmouth.edu/ncsu/mobilitymodels, July, 2009.

6.2.3. System parameters

The system parameters is the checkpoint period N and the re-clustering threshold θ. We set $N \in \{10, 20, 40, 60\}$ referring to a certain number of time units. For synthetic traces, N refers to values from 0.16 min to 1 min, while for the real traces, N refers to values from 1.67 min to 10 min. We set $\theta \in \{0.3, 0.4, \ldots, 0.7\}$.

Fig. 7 demonstrates the impact of θ on the (average) cohesion metric and the obtained ratio α for the T0 trace. Similar results are obtained for the other traces. A low θ value results to extension of the interval between consecutive global re-clusterings, since the (average) group density is allowed to be small enough compared to the initial group density. A decrease in the frequency of global re-clusterings leads to a decrease in the induced communication overhead. Moreover, the cohesion is high with low θ, which derives from low group density values. Once the number of members per group is low then the probability that a group is fragmented is lower than the fragmentation probability of a group with many members. In the experiments we set $\theta = 0.3$ in order to achieve high cohesion and low α.

Fig. 8 shows the MPD of groups being in the OS state against N for the synthetic and real traces. For the synthetic traces the percentage of time in the OS state is very small (less than 0.5% and 2%, respectively). This means that the OGV stops shortly after the group formation and the group persists for a relatively long time interval. For the real traces, OGV delays the conclusion for group persistence up to 17% in order to increase its confidence.

6.3. Performance assessment

6.3.1. OGV optimality

In order to demonstrate the optimality achieved by the OGV, we experiment with a GVS_{t_s}, which takes a validation decision at a deterministic (fixed) stopping time t_s and compare its performance with the OGV, which stops at t_*. Let the ratio $\frac{\gamma}{\alpha}$ denote the normalized information accuracy against load ratio. A high $\frac{\gamma}{\alpha}$ value indicates that GVS achieves high accuracy with low communication cost. Fig. 9 shows the reward Y_t (left) and $\frac{\gamma}{\alpha}$ (right) obtained from (i) OGV, which stops at t_* with $MPD = t_*/(t_* + t_\#)$ and (ii) GVS_{t_s}, which stops at t_s with $MPD = t_s/(t_s + t_\#) \in \{0\%, \ldots, 60\%\}$.

Remind that the reward Y_t represents the group coherency at the stopping time t_*. The comparison of the proposed model with a deterministic one, for the discussed metrics, reveals that our model outperforms the deterministic leading to a more coherent group of nodes saving communication cost and maintaining the information accuracy at high levels. By adopting the two performance metrics, we try to reveal the efficiency of our model concerning the high group coherency (OST related performance metric) and information quality accompanied by the low communication cost (network related performance metric).

The peak of each curve (i.e., maximum reward and maximum ratio) in Fig. 9 corresponds to $t_*/(t_* + t_\#)$, which corresponds to the optimal stopping time t_* achieved by the OGV. This indicates the optimality obtained by the proposed OGV compared to any fixed stopping policy. Note that for $t_s = t_f$ we obtain a GVS which immediately treats a group as a persisting group.

6.3.2. OGV performance

Table 4 shows the mean accuracy γ for all groups against N, the mean number of re-clustering events w out of trace length T, and the portion f of the members of the fragmented groups in $\mathcal{F}$ out of the total number of nodes $|V|$.

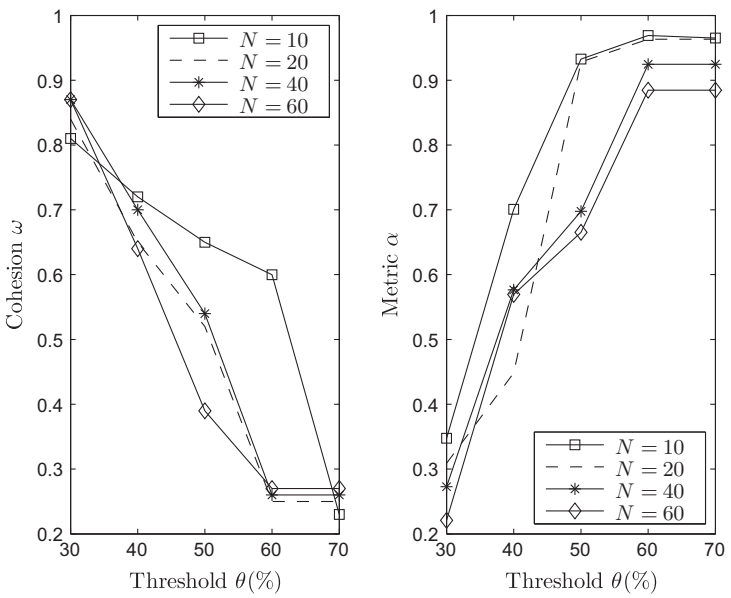


Fig. 7. The cohesion ω and ratio α vs. re-clustering threshold θ.

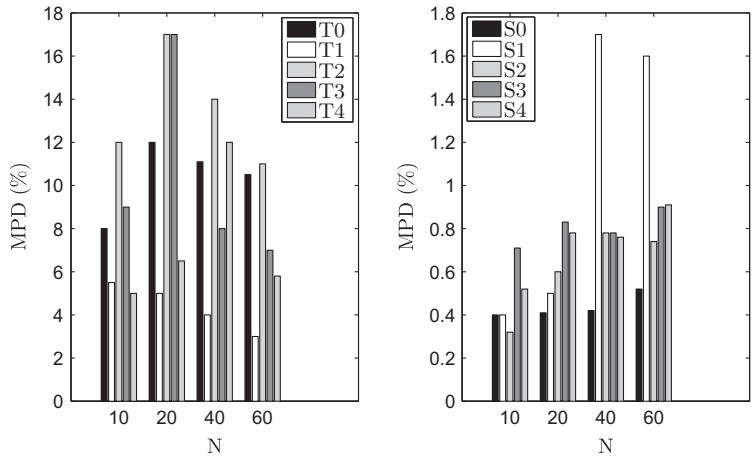


Fig. 8. The MPD vs. checkpoint period N.

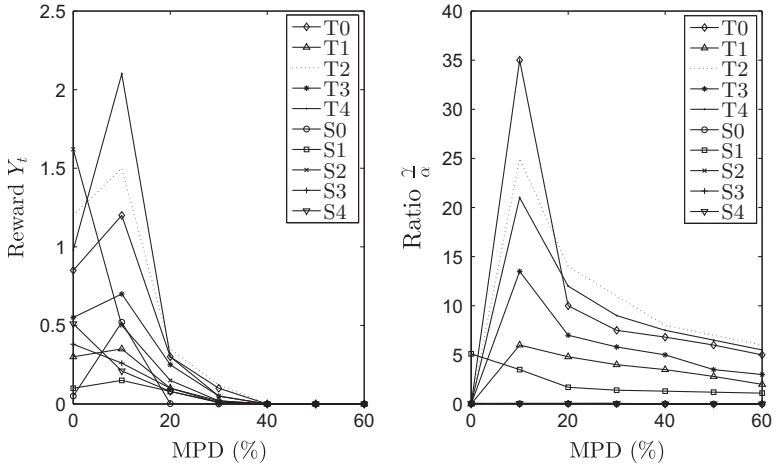


Fig. 9. The reward Y_t and ratio $\frac{\gamma}{\alpha}$ vs. MPD; $N = 40$.

Table 4
The γ, w, and f metrics.

Trace	T0	T1	T2	T3	T4	S0	S1	S2	S3	S4
γ										
$N = 10$	.988	.988	.998	.987	.988	.985	.991	.983	.981	.949
$N = 20$	.987	.987	.998	.986	.987	.981	.971	.979	.990	.973
$N = 40$	.987	.983	.987	.978	.987	.974	.968	.965	.970	.907
$N = 60$	.986	.977	.987	.977	.981	.974	.958	.952	.892	.878
$w(\%)$	2.76	1.72	3.43	0.88	2.77	0	0.02	0.07	0.09	0.05
f	0.238	0.244	0.279	0.196	0.144	0.1	0.14	0.13	0.16	0.15

It is worth noting that for all traces, we obtain high accuracy, which decreases minimally as N increases. This signifies the applicability of the OGV to location-aware applications. In addition, the mean number of re-clustering events is relatively low for all traces (note that w is up to 3.43%). Interestingly, although f is relatively high for some of real traces (e.g., T0, T1, T2) the corresponding $w(\%)$ is low. Through the adaptation mechanism of β, the OGV maintains the number of re-clustering events low. This means that the OGVs for the (formed) sub-groups, which are derived from the fragmented groups, prolong the corresponding validation decision, thus, avoiding further fragmentations and, consequently, re-clustering events. This indicates an advantage of the proposed approach, which minimizes the number of $GFA(V)$ invocations.

Fig. 10 shows the δ metric against N. One can observe that δ scales well with the checking periods. For the majority of the synthetic traces we obtain very low δ ($\leqslant$0.5%).

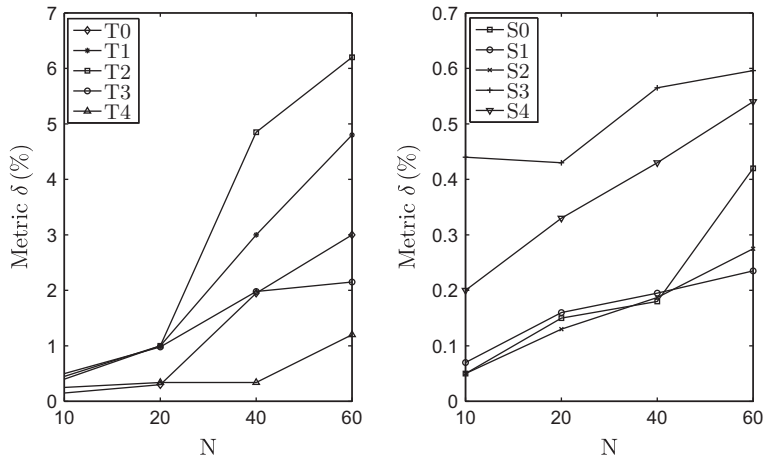


Fig. 10. The δ metric vs. checkpoint period N.

This is not the case for traces S3 and S4 especially when $N \geqslant 40$. However, all the synthetic traces achieve $\delta < 0.6\%$ no matter the N value. For the real traces, a relatively high f (see Table 4 for T0, T1, and T2) correlates with relatively high δ as N increases. A long checkpoint period indicates fewer (sparse) checkpoints. Hence, the system delays in detecting groups that may turn fragmented for periods of time (i.e., $t_l - t_d$) shorter than the interval between checkpoints. In such cases, δ is relatively high. On the other hand, if we increase the frequency of checkpoints (i.e., adopt low N values) then the duration $t_l - t_d$ decreases, thus, leading to low δ.

Figs. 11 and 12 show the ratio α and gain g against N. The α and g values demonstrate relatively limited fluctuation with respect to N; $\alpha \pm 15\%, g \pm 15\%$. A long checkpoint period results to low communication overhead, however, at the expense of high δ. For the synthetic traces we obtain significantly low communication overhead (90% less communication load), thus, relatively high gain. Regarding the real traces, we obtain 55% less communication load, on average. OGV achieves an average gain close to 0.8 and 0.6 for synthetic and real traces, respectively. This justifies the adoption of the proposed scheme by a LBS system for decreasing the induced vertical communication load, while maintaining high accuracy.

The above discussed values for our performance metrics are calculated as the average of our results for a high number of simulations. For completeness, we give an indication of the confidence interval for our results by presenting the marginal error for the interval of each parameter. We conduct a large number of experiments for each trace and get the average margin of error of each parameter. Accordingly, for the entire set of traces, we calculate the final average margin of error. It should be noted that we adopt the typical formula for calculating the margin of error as provided by statistics and consider a confidence level of 95%. For the entire set of our simulations and the entire set of our synthetic traces the average margin of error is as follows: (i) for Y_t is equal to 0.17, (ii) for *MPD* is equal to 0.09 and, (ii) for γ is equal to 0.13.

6.3.3. Adaptive OGV performance

In order to demonstrate the effectiveness of the adaptation mechanism, we study the performance of a OGV with constant β. Table 5 shows the average MPD(%) and average $\alpha(\%)$ metrics over all N with constant $\beta \in \{0.1, 0.5, 0.8, 0.9\}$. Low β results to low MPD, thus, low communication load. On the other hand, high β has as result a MPD of 47% (on average of all traces) with high communication load. Evidently, as $\beta \to 1$, the system concludes on the group persistence later in time. Therefore, low β would be preferable. This, however, is not reasonable. Low β results to rapid conclusions on group persistence, thus, leading to 'pre-mature' decisions. As a consequence, groups, which might be 'thoughtlessly' classified as persisting groups, get fragmented and then a relatively high number of re-clustering events occurs. Table 6 shows the gain g and the portion of re-clustering events w for $N \in \{40, 60\}$. The entries typeset in boldface corresponds to $N = 60$. We observe that as $\beta \to 1$ then $w \to 0$ (a few re-clusterings occur) but $g \to 0$ (low gain is attained). The adaptation mechanism attempts to balance the induced communication load and the number of re-clustering events.

6.4. Comparative assessment

We perform the comparative assessment of our scheme against the (continuous) threshold-based K-means monitoring model (TKM) in [13] and the simple reference solution (REF) discussed also in [13]. In the comparative assessment we experiment with the total communication load c and the percentage of re-clustering events $w(\%)$. We redeveloped the TKM and REF models in the Java programming language for the comparison.

6.4.1. The REF model

The K-means algorithm divides a set V of nodes into a set $\mathcal{Q}$ of K groups/clusters such that the centroid distance of the groups is minimized. The REF model in [13] has as follows. At time $t = 0$, every node u reports its position $\mathbf{p}_u$ to the system (uplink signalling messages). The system

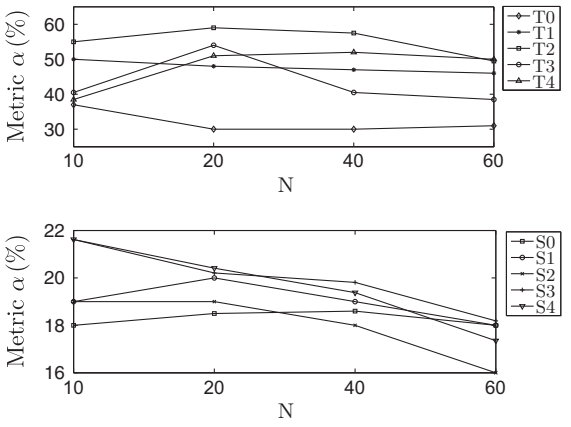


Fig. 11. The ratio $\alpha(\%)$ vs. checkpoint period N.

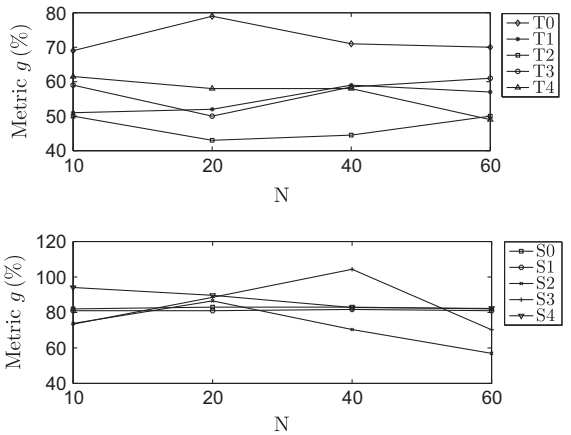


Fig. 12. The gain $g(\%)$ vs. checkpoint period N.

Table 5
The MPD and ratio α for constant β.

β	MDP (%)					$\alpha(\%)$				
	T0	T1	T2	T3	T4	T0	T1	T2	T3	T4
0.1	3	6	9	6	4	23	48	50	42	46
0.5	6	9	16	11	11	25	53	54	44	54
0.8	33	32	47	27	30	93	94	92	84	63
0.9	41	41	54	63	40	99	95	95	91	77

performs a K-means GFA ([58]) and produces the $\mathcal{Q}_0$. At every $t > 0$, whenever a node moves, it sends a location update to the system. The system obtains the new $\mathcal{Q}_t$ over the updated locations through the invocation of the K-means GFA using the K centres from $\mathcal{Q}_{t-1}$ as the seeds [13].

It is worth noting that we could adopt an alternative static GFA, like the fuzzy K-means clustering algorithm (FKM) as a reference model. However, FKM has computational complexity $O(K^2|V|r)$ while REF requires $O(K|V|r)$, where r is the numbers of iterations needed to converge [61]. Hence, we obtain higher computational load than the considered REF especially when we require a high

Table 6
The gain g and w metrics for constant $\beta; N = (40, \mathbf{60})$.

β	T0	T1	T2	T3	T4
Gain $g(\%)$					
0.1	78, **85**	50, **52**	50, **53**	61, **60**	51, **50**
0.5	73, **80**	50, **51**	49, **50**	58, **60**	50, **50**
0.8	10, **12**	11, **11**	40, **47**	22, **25**	20, **20**
0.9	9, **10**	11, **10**	22, **20**	12, **11**	11, **11**
Re-clustering events percentage $w(\%)$					
0.1	6.2, **0.6**	1, **0.5**	2.5, **2.8**	0.8, **0.05**	2, 2.1
0.5	6.1, **0.5**	1, **0.5**	2.5, **2.8**	0.8, **0.05**	1.5, 2
0.8	0.1, **0.1**	0.05, **0.05**	2.2, **2.5**	0.05, **0.05**	0.05, **0.05**
0.9	0, **0**	0, 0	2, **1.8**	0, **0**	0, **0**

Table 7
Comparison parameters.

Notation	Value/range
Model TKM	
Tolerance Δ (in meters)	10, 20
Spatial boundary thr. a	0.7
Model REF	
Number of groups K	2, 8, 10, 12
Model OGV	
Re-clustering thr. θ	0.3, 0.5
Checkpoint period N	10, 20, 40, 60

number of formed groups. Moreover, we do not achieve any better performance by adopting the FKM instead of REF, since in our case FKM also needs to define a priori a fixed number of groups. The limitations and unsuitability of static GFAs are discussed on Section 2.

6.4.2. The TKM model

The TKM initially specifies a tolerance Δ and a number of groups K. At time $t = 0$, the set $\mathcal{Q}_0$ is the same as in REF. Then the system disseminates (downlink signalling messages) to every node u a (spatial boundary) threshold $\psi_u = a\Delta, a \in (0, 1)$. At $t > 0$, a node u reports its location to the system (uplink signalling message) only if its current position deviates from the previous one by at least ψ_u. When the system receives such report, it obtains the new set $\mathcal{Q}$ over the last recorded positions of the other nodes (which are still within their assigned thresholds) through the invocation of the K-means GFA.[3] Consequently, the system computes new thresholds and sends them (downlink signalling messages) to a fraction of the nodes (or, in the worst case, to all nodes). In REF all messages refer to uplink signalling messages, i.e., location updates from nodes to the system. TKM includes also downlink signalling messages, through which the system informs the nodes about their new thresholds. We assume that the cost of uplink and downlink signalling messages are equal as in [13].

6.5. Model comparison

We measure the total communication load c, i.e., uplink and downlink signalling messages between the system and

[3] The authors in [13] propose an improved *hill climbing* method for the K-means GFA.

nodes, for the REF, TKM and OGV models for each real trace. The values of $N \in \{10, 20, 40, 60\}$ and $\theta \in \{0.3, 0.5\}$ are set for the OGV. The number of groups $K \in \{2, 8, 10, 12\}$ and the tolerance $\Delta \in \{10, 20\}$ with $a = 0.7$ as in [13] for REF and TKM models. The Δ values are in meters. Table 7 summarizes the setting of the comparison parameters.

Figs. 13–17 illustrate the total number of messages as a function of N and θ, for traces T0–T4, respectively. For each trace, REF and TKM do not involve N and θ, thus, their corresponding results are constant and illustrated for comparison reasons. The load for TKM is the average load obtained for all K. The load for the OGV is the average load obtained for all N when experimenting with diverse θ values, while when experimenting with diverse N values, we set $\theta = 0.3$.

Fig. 13(left) demonstrates the effect of N on the load, where the OGV assumes the lower load in comparison with the other models; (58% and 77% less messages than TKM and REF, respectively). In the case of the OGV, when N is large, uplink signalling messages are sent with low frequency from nodes to the system. In the case of TKM, when Δ is large, nodes may not be in need of being updated by a downlink signalling message. That is because, at some time instances, there are no location updates, and therefore, no threshold dissemination is required [13]. However, in this case ($\Delta = 20$), TKM needs 42% more messages than the OGV in order to maintain updated clustering information. Fig. 13(right) shows the impact of θ on the average load for the OGV, corresponding to all N values. When θ is low, a few re-clustering events occur in our model. Consequently, we obtain a few uplink signalling messages for location reports. In this case, the OGV achieves 50% (74%) less load than TKM (REF). When θ is large, that is the re-clustering events occurs more often, then the OGV involves more uplink signalling messages due to global re-clustering. Nevertheless, in this case, for $\theta = 0.5$ the OGV needs 30% (61%) less messages than TKM (REF). Similar results are obtained for the other traces in Figs. 14–17.

The last experiment studies the percentage of (global) re-clustering shown in Table 8 for TKM, REF (average values for all K) and in Table 4 for the OGV (with $\theta = 0.3$). REF performs re-clustering almost surely at each time instance. TKM performs re-clustering only when there is a threshold violation [13]. The OGV requires re-clustering based only on the current group density. One can observe that the OGV achieves very low rate of re-clustering compared to TKM (and REF). The OGV outperforms TKM and REF by a wide margin on computational cost on the LBS server (due to clustering re-evaluation) and communication overhead. Summarizing the comparative assessment, in all real traces, the OGV achieves more than 60% (and sometimes close to 80%) reduction on the overall vertical communication overhead than TKM and REF, respectively. The OGV incurs significantly fewer uplink signalling messages since groups are considered persistent at early stages of group validation. In addition, the number of downlink signalling messages in the OGV never exceeds the number of uplink signalling messages, since the system disseminates once a downlink signalling message to a group in its entire lifetime.

7. Discussion on OGV performance

In this section, we discuss possible load savings obtained from the OGV application w.r.t. application data flow from system to set of mobile users V. Let us call this data flow as downlink data messages. Location-sensitive content is pushed by the system to mobile user u, e.g., proximity-based notification service which pushes data when u crosses/enters a pre-defined spatial boundary/region $\mathcal{A}_u \subset \mathbb{R}^2$. Consider the event 'user u enters/crosses a region/boundary $\mathcal{A}_u$'. The event definition is application dependent, i.e., on $\mathcal{A}_u$. The realization of such event depends on the mobility behaviour $\mathcal{M}_u$ of user u. Such event occurs with average frequency z_u being a function of $\mathcal{A}_u$ and $\mathcal{M}_u$. The quotient $\frac{1}{z_u}$ denotes the average number of downlink data messages. The $\mathbb{E}(z_u)$ denotes the average frequency of all $u \in V$. In the conventional scenario, the average application communication load $c_{app.}$ is

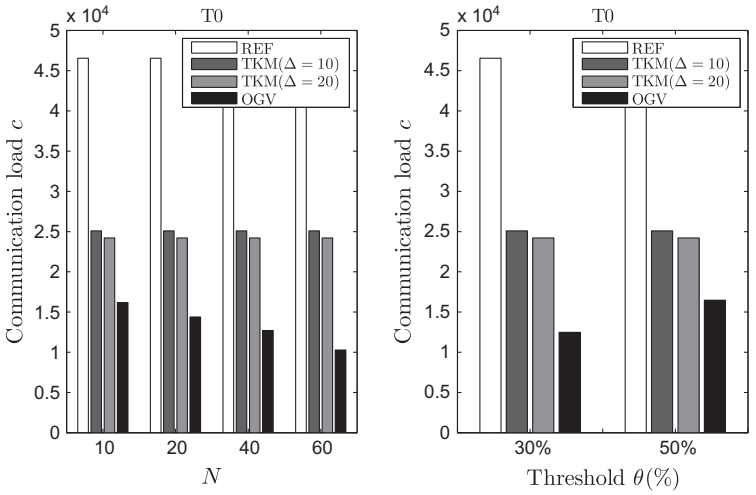


Fig. 13. Communication load vs. checkpoint period N(left) and vs. re-clustering threshold θ(right) for trace $T0$.

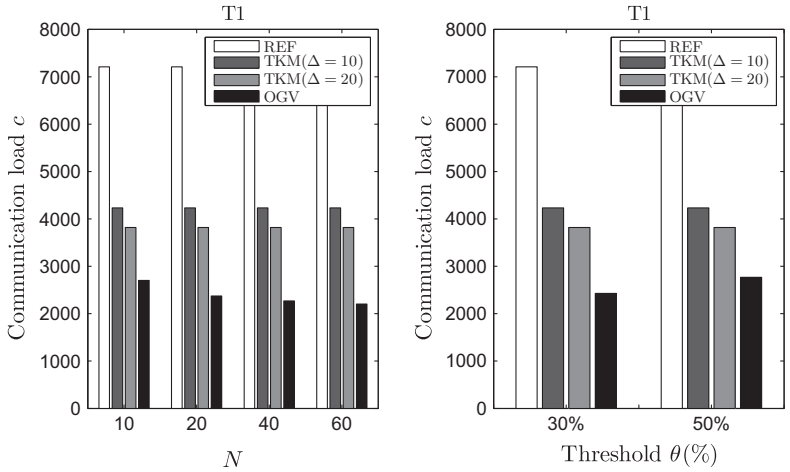


Fig. 14. Communication load vs. checkpoint period N(left) and vs. re-clustering threshold θ(right) for trace $T1$.

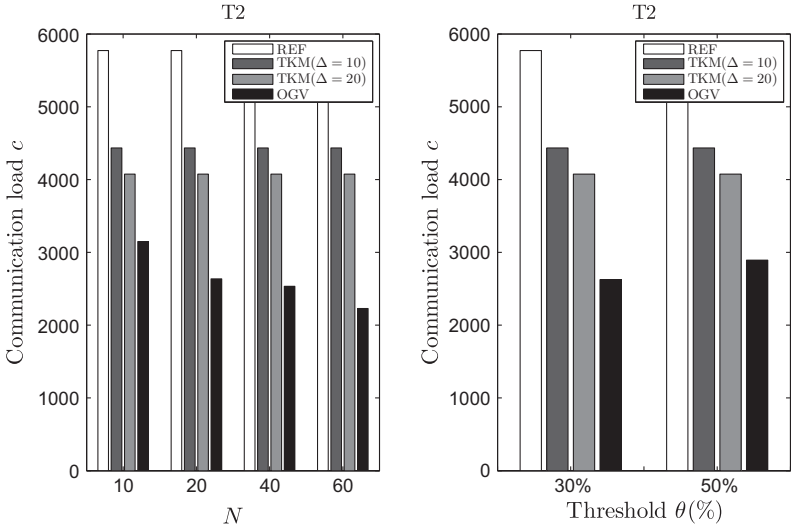


Fig. 15. Communication load vs. checkpoint period N(left) and vs. re-clustering threshold θ(right) for trace $T2$.

proportional to $|V|\mathbb{E}(z_u)$. If $|\mathcal{Q}| \leqslant |V|$ is the number of groups derived from OGV, which also denotes the number of GLs, then the average application communication load in our case (in which GL is responsible to disseminate application data to group members) is $c_{app.}^{OGV} \propto \mathbb{E}(|\mathcal{Q}|)\mathbb{E}(z_u)$, where $\mathbb{E}(|\mathcal{Q}|)$ is average number of groups.

The obtained gain $g_{app.}$ in terms of application load is proportional to $\mathbb{E}(z_u)(|V| - \mathbb{E}(|\mathcal{Q}|))$, which is $O(\mathbb{E}(z_u)|V|)$. That is as we exploit the derived groups and $\mathbb{E}(|\mathcal{Q}|) \ll |V|$ then gain is quite significant. Moreover, as $\mathbb{E}(z_u)$ is high thus application content updates occur with high frequency, then our scheme is deemed appropriate for adoption. The upper bound of $g_{app.}$ is $(|V| - 1)\mathbb{E}(z_u)$ (in the extreme case where all nodes are clustered in one group), while the lower bound is zero, i.e., when all groups are singletons. Moreover, GL is responsible for disseminating (local broadcast or unicast) application data to group members. In local broadcast, GL disseminates one message to all members, while in local unicast, GL disseminates $|Q| - 1$ messages. Such information might be unreachable for some time instances for a fraction of members. This is estimated by γ metric. There is a trade-off between obtained gain $g_{app.}$ and induced accuracy γ. The normalized $c_{app.}$ out of induced accuracy in a conventional scenario is $|V|\mathbb{E}(z_u))/\gamma$ with $\gamma = 1$. In our case, we obtain the normalized $c_{app.}^{OGV}$ is $\mathbb{E}(|\mathcal{Q}|)\mathbb{E}(z_u)/\gamma$, with $\gamma \leqslant 1$. We obtain a certain load benefit when $c_{app.}^{OGV} < c_{app.}$ or $\mathbb{E}(|\mathcal{Q}|)\mathbb{E}(z_u)/\gamma < |V|\mathbb{E}(z_u)$ or $\mathbb{E}(|\mathcal{Q}|) < \gamma|V|$. In our simulations, average $\bar{\gamma} = 0.95$. This indicates that even when the average cluster density is very low the gains stemming from the proposed scheme are quite important.

In short, the strengths of our model are as follows:

- We provide a fast and incremental GFA without predetermined groups, which are initially unknown and be scalable in the number of nodes.

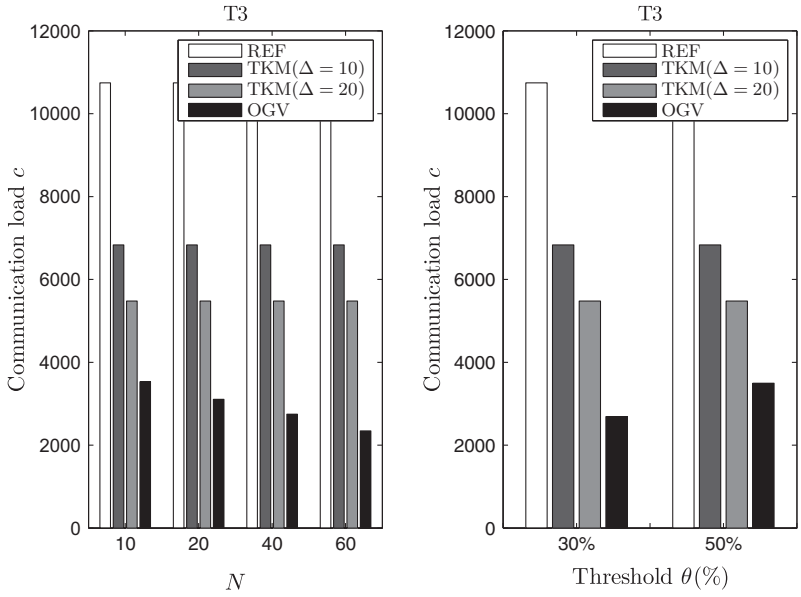


Fig. 16. Communication load vs. checkpoint period N(left) and vs. re-clustering threshold θ(right) for trace $T3$.

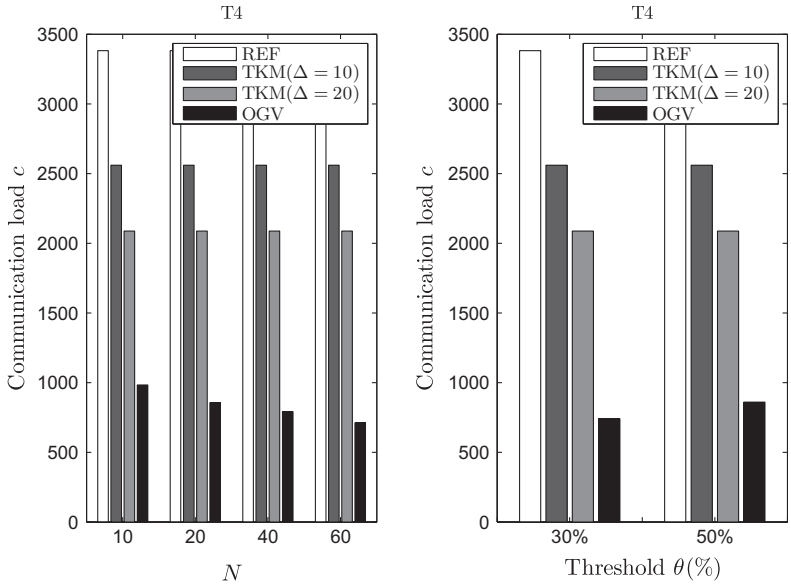


Fig. 17. Communication load vs. checkpoint period N(left) and vs. re-clustering threshold θ(right) for trace $T4$.

- We provide a time optimized decision mechanism on the optimal time to stop monitoring all nodes and communicate only with the GL.
- We propose an on-line algorithm for decision making, thus, being applicable to mobile applications.
- We adopt periodic determination of GLs (a node will not bea GL for ever) and two GL substitution methods.

On the other hand, the following list reports the limitations of our model:

- Our scenario does not provide an incentive mechanism for acting a node as a GL.
- The incremental GFA is fast, however, it does not guarantee the optimal clustering (there is a trade-off between the optimal group partitioning and the computational complexity).
- The cohesion metric does not take into account the position correlation between mobile nodes.

8. Application domains

We argue that our scheme has numerous practical applications in the LBS domain. For instance, in real-time traffic control systems, efficient monitoring of groups of vehicles helps detecting congestions [5]. LBSs distribute

Table 8
The percentage of re-clustering events.

$w(\%)$	T0	T1	T2	T3	T4
REF	98	97	97	99	99
TKM ($\Delta = 10$)	58	60	78	59	77
TKM ($\Delta = 20$)	55	60	70	47	71

on-demand information to groups of vehicles describing road conditions and available facilities in some geographic area, e.g., traffic accident alerts [7]. Mobile advertising LBS delivers advertisements to group representatives, who in turn share content with other users around them [8]. The push-based LBS in [9] delivers information from a campus LBS server to groups of students according to their spatio-temporal context (postponed lessons). The public transportation information system of the Taipei city[4] delivers updates, pertinent to the arrival time of buses, to groups of waiting passengers.

Consider the following scenario, which emerges in evacuation planning/control systems [11]. Imagine a risk, like fire, occurring in a large building or indoor facility, e.g., museum, airport. The system monitors each individual's position and upon risk occurrence transmits evacuation instructions to visitor-carried mobile devices (e.g., smartphones, venue portable devices). The system identifies groups of visitors and transmits evacuation instructions to certain group members to save network resources which may be rapidly and uncontrollably exhausted under emergency conditions.

Consider the following scenario, which emerges in the case of networked museum portable guides [10] (or some other venue of similar use). We assume that such guides retrieve multimedia content through the underlying network and push it to the requesting visitor. The system needs to know the location of the visitor in order to determine the part of the underlying database that should be provided to him/her, e.g., multimedia presentation on an exhibit. Through the proposed mechanism the system detects the persistence of a group of visitors, determines the location of a group representative (e.g., GL, guide) and transmits location-dependent information only to him/her. Such data are passed to the rest members of the group through a short range communications component.

9. Conclusions

This paper proposes a system for time-optimized validation of groups of moving objects in Location Based Services. An incremental Group Formation Algorithm (GFA) is proposed and the Optimally Scheduled Group Validation System (OGV) is introduced based on the Optimal Stopping Theory. We present an optimal stopping policy for each formed group through the OGV by exploiting the group formation information: compactness and coherency. We assess the proposed system with real and synthetic traces and compare OGV with other models found in the literature w.r.t. communication load and re-clusterings on the LBS server using real traces. We conclude that OGV achieves considerable network and computation savings, by decreasing the vertical communication load keeping information accuracy at high levels.

The OGV engages a period for checking the validity of a formed group, while it is in the CF state. This period is fixed and affects the communication cost from all group members, since they have to report their position to the system in order to check the group persistence. However, OGV can be extended to more sophisticated adaptation mechanism of the checkpoint period N. The checkpoint period N can be tuned according to the network or computation load seen at the system or the mobility behaviour of the groups. Intuitively, once a group exhibits a persisting behaviour that lasts for a period greater than N, then N could be increased, thus, avoiding extra communication cost due to the group members' location report at each checkpoint. One the other hand, if a group exhibits a rather unstable persisting behaviour then we can increase the checkpoints rate in order to evaluate the persistence of the group. Moreover, the scheme can be enhanced with a mechanism which manages small changes in group membership in a smoother way, i.e., a single group 'defector' does not trigger the invalidation of the entire group but its dynamic restructuring. In this case, the provisions for group validation need to be modified.

Appendix A

In this appendix we study the characteristics of the rewards Y_t under the filtration $\mathbb{F}_t = \mathbb{B}(\Omega_1, \ldots, \Omega_t)$ referring to the Lemmas 2–4 and Theorem 1 in [44]. Then, we provide the optimal stopping criterion in our case. Let $S_t = \Omega_1 + \cdots + \Omega_t$ and $\beta^t = \prod_{k=1}^{t} \beta_k$ with the conventions that $S_0 = 0$ and $\beta^0 = 1$. In our case, Ω_t are non-negative and $\beta_k = \beta \in (0, 1)$ constant. We assume that at $t = 0$ the OGV initiates the validation process with some $\omega \in [0, 1]$ (i.e., initial state of the validation process). Then, we define the function

$$J(\omega) = \sup_t \mathbb{E}(\beta^t(\omega + S_t))$$

and t ranges over all stopping times for $\Omega_1, \Omega_2, \ldots$. The problem is to find an optimal t_* for initial state ω, that is a t_* such that the $\sup_t \mathbb{E}(\beta^t(\omega + S_t))$ is attained. Interestingly, this problem has a solution given in [48] (the 'burglar problem') in which t_* is evaluated by adopting the principle of optimality [59] provided that: (i) Ω_t are non-negative, (ii) $\omega = 0$ at $t = 0$, and (iii) β is constant. This is our case.

Now, from Lemma 2 [44], $J(\omega)$ is convex and nondecreasing; i.e., there exists a unique number $s > 0$ such that $\omega < J(\omega) < s$ for $\omega < s$ and, for $\omega \geqslant s, J(\omega) = \omega$. Then, the application of the principle of optimality to such function denotes that $J(\omega) \geqslant \mathbb{E}(\beta J(\omega + \Omega))$ for all ω; (Lemma 3 [44]). Hence, the process: $J(\omega), \beta^1 J(\omega + S_1), \beta^2 J(\omega + S_2), \ldots$ is a nonnegative, expectation decreasing semimartingale. Through this, $J(\omega) \geqslant \mathbb{E}(\beta^t(\omega + S_t))$ for all stopping times t (Lemma 4 [44]).

[4] http://english.dot.taipei.gov.tw/.

For each $z \geqslant 0$, let $\tau(z)$ be the first stopping time t such that $S_t > z$. For $\omega \leqslant s$, let $r(t,\omega) = \min(t, \tau(s-\omega))$. Based on Theorem 1 [44] the process $\{\beta^{r(t,\omega)} J(\omega + S_{r(t,\omega)}), t = 0, 1, \ldots\}$ is a uniformly integrable martingale and converges certainly to $\beta^{\tau(s-\omega)} J(\omega + S_{\tau(s-\omega)})$ since J is continuous and $r(t,\omega)$ converges to $\tau(s-\omega)$. In addition, for each $\omega \leqslant s, \tau(s-\omega)$ is optimal for ω or, equivalently, $J(\omega) = \mathbb{E}(\beta^{\tau(s-\omega)}(\omega + S_{\tau(s-\omega)}))$ (Theorem 1 [44]).

For $\omega = 0$, we obtain $J(0) = \mathbb{E}(\beta^{\tau(s)} S_{\tau(s)}) = \mathbb{E} Y_{\tau(s)}$ and s is obtained from $s = \mathbb{E}(\beta^{\tau(0)}(s + S_{\tau(0)}))$ or $s = \mathbb{E}(\beta^{\tau(0)} S_{\tau(0)})/(1 - \mathbb{E}\beta^{\tau(0)})$. The determination of the expectation of $\beta^{\tau(0)}$ and $S_{\tau(0)}$ has as follows: $\tau(0)$ is the minimum t such that $S_t > 0$. Since Ω is non-negative, thus, one sees that the problem is monotone, then $\tau(0) \equiv 1$ and $S_{\tau(0)} \equiv \Omega_1$. That is because, the reward for stopping s must be the same as the reward for continuing using the rule that stops the first time the sum of the future observations is positive. Hence, in our case, we obtain the optimal stopping criterion $s = \frac{\beta}{1-\beta}\mathbb{E}\Omega$, thus, from the one-stage look-ahead optimal rule [50], OGV stops at the first time t at which $S_t \geqslant s$.

Appendix B

Given the sequence of $\omega_1, \omega_2, \ldots, \omega_t$, the probability density estimation up to t is $\hat{f}_t(\omega; (\omega_1, \ldots, \omega_t)) = \hat{f}_t(\omega; \omega^t)$ where $\omega^t = (\omega_1, \omega_2, \ldots, \omega_t)$. The kernel method is a widely adopted for non-parametric density estimation method, thus, the $\hat{f}_t(\omega)$ based on t cohesion values $\boldsymbol{\omega}^t$ is

$$\hat{f}_t(\omega) = \frac{1}{t}\sum_{k=1}^{t} K_h(\omega - \omega_k)$$

with $K_h(u)$ is a kernel function, which is a unimodal, symmetric, non-negative function that centres at zero and integrates to unity. The *window* h controls the degree of smoothing of the estimation. An optimal selection for h is provided by [60]. The expected value of Ω based on $\boldsymbol{\omega}^t$, $\mathbb{E}(\Omega; \boldsymbol{\omega}^t)$ is calculated by $\int_0^1 \omega \hat{f}_t(\omega)\,\mathrm{d}\omega$. Interestingly, we can incrementally evaluate the $\hat{f}_t(\omega)$, that is, calculating $\hat{f}_t(\omega)$ based on $\hat{f}_{t-1}(\omega)$ and the observed cohesion ω_t at $t > 1$. We estimate the density function in the following incremental manner:

$$\begin{aligned}\hat{f}_t(\omega) &= \frac{1}{t}\sum_{k=1}^{t} K_h(\omega - \omega_k)\\ &= \frac{1}{t}\left(\sum_{k=1}^{t-1} K_h(\omega - \omega_k) + K_h(\omega - \omega_t)\right)\\ &= \frac{t-1}{t}\hat{f}_{t-1}(\omega) + \frac{1}{t}K_h(\omega - \omega_t)\end{aligned}$$

with $\hat{f}_1(\omega) = K_h(\omega - \omega_1)$. Hence, for the expected value $\mathbb{E}(\Omega; \boldsymbol{\omega}^t)$, we obtain:

$$\mathbb{E}(\Omega; \boldsymbol{\omega}^t) = \frac{t-1}{t}\mathbb{E}(\Omega; \boldsymbol{\omega}^{t-1}) + \frac{1}{t}\int_0^1 \omega K_h(\omega - \omega_t)\,\mathrm{d}\omega$$

When ω_t is received, only the evaluation of $g_t(\omega_t) = \frac{1}{t}\int_0^1 \omega K_h(\omega - \omega_t)\,\mathrm{d}\omega$ is needed, with $\mathbb{E}(\Omega; \boldsymbol{\omega}^1) = g_1(\omega_1)$. We adopt the Gaussian kernel function $K_h(u) = \frac{1}{\sqrt{2\pi}h} e^{-\frac{1}{2}\left(\frac{u}{h}\right)^2}$ and, thus,

$$\begin{aligned}g_t(\omega) &= \frac{1}{\sqrt{2\pi}ht}\int_0^1 u e^{-\frac{1}{2}\left(\frac{u-\omega}{h}\right)^2}\,\mathrm{d}u\\ &= \frac{1}{\sqrt{2\pi}ht}1.253\omega h\left(erf\left(\frac{\sqrt{2}}{2h}\omega\right) - erf\left(\frac{\sqrt{2}}{2h}(\omega - 1)\right)\right)\\ &\quad + h^2\left(e^{-0.5\left(\frac{\omega}{h}\right)^2} - e^{-0.5\left(\frac{\omega-1}{h}\right)^2}\right)\end{aligned}$$

with $erf(u) = \frac{2}{\sqrt{\pi}}\int_0^u e^{-z^2}\,\mathrm{d}z$ (the error function).

Appendix C

In this appendix we discuss a solution of the optimal stopping problem (see Section 4.3) in which the validity values are linearly correlated. Consider that $\omega_t = \mu\omega_{t-1} + o_{t-1}$, for $t = 1, 2, \ldots$, with $\mu \in [0, 1)$, O_t are i.i.d. random variables with realization o_t and known probability distribution, and $\omega_0 = 0$. It can be easily shown that $\omega_t = \sum_{k=0}^{t-1}\mu^{t-1-k}o_k$. The cumulative sum of validity values up to t is then $S_t = \sum_{k=1}^{t}\sum_{i=0}^{k-1}\mu^{k-1-i}o_i$. Now, consider that $S_t = s$ and it is optimal to stop the validation process (i.e., take decision D1). Then, the reward is $Y_t = \beta^t S_t = \beta^t s$ and is, at least, as large as any expected future reward $\mathbb{E}\{\beta^{t+\tau}(s + S_\tau)\}$, i.e., $s \geqslant \frac{\mathbb{E}\{\beta^\tau S_\tau\}}{1 - \mathbb{E}\{\beta^\tau\}}$ for all stopping times τ. The same must hold true for all $s' \geqslant s$ so that the optimal stopping time t_* is of the form $t_* = \min\{t \geqslant 1 | S_t \geqslant s_0\}$, for some s_0. The reward for D1, i.e., s_0, must be the same as the reward for continuing (i.e., D2) using the rule that stops the first time the sum S_τ is positive (see Appendix A). That is, s_0 must satisfy the equation $s_0 = \mathbb{E}\{\beta^\tau(s_0 + S_\tau)\}$. Since Ω is non-negative with probability one, then $\tau \equiv 1$ and $S_\tau \equiv \Omega_1$ ([44]), and then s_0 has the form $s_0 = \frac{\beta}{1-\beta}\mathbb{E}\Omega$. Hence, OGV stops the validation process at

$$t_* = \min\left\{t \geqslant 1 | \sum_{k=1}^{t}\sum_{i=0}^{k-1}\mu^{k-1-i}o_i \geqslant s_0\right\}.$$

Appendix D

In this appendix we quantify the robustness of the proposed scheme in light of node failures. We concentrate our study on the GL and deputies as introduced in Section 5.5.1, 'soft' GL substitution. Consider the set Q_ϕ of the deputies of a group Q which corresponds to radius ϕR. Table 9 shows the average cardinality of the set Q_ϕ, $|Q|_\phi$, for all traces with $\phi \in \{0.25, 0.33, 0.5\}$.

Let $P_\phi = 1 - (1-p)^{|Q|_\phi}$ indicate the probability at least one deputy from Q_ϕ is alive (operational after some time from initial assignment within the range ϕR from the current GL) with probability $p \in \{0.7, 0.8, 0.9\}$; $1 - p$ denotes the failure probability of a single node. That is, P_ϕ is the probability that Q has a GL based on the 'soft' GL substitution strategy. Table 10 shows the probabilities

Table 9
Average $|Q|_\phi$ against ϕ.

ϕ	S0	S1	T0	T1	T2	T3	T4
0.25	6.3	5.7	2.1	4.1	1.3	1.4	2.7
0.33	7.7	7.1	2.4	5.1	1.5	1.5	3.4
0.5	9.3	8.7	2.9	6.7	1.6	1.8	4.6

Table 10
The P_ϕ^c and P_ϕ^f.

Trace	$p = 0.7$		$p = 0.8$		$p = 0.9$	
	P_ϕ^f	P_ϕ^c	P_ϕ^f	P_ϕ^c	P_ϕ^f	P_ϕ^c
S0	0.99	0.99	1.0	1.0	1.0	1.0
S1	0.99	0.99	0.99	1.0	1.0	1.0
T0	0.91	0.97	0.96	0.99	0.99	0.99
T1	0.99	0.99	0.99	0.99	1.0	1.0
T2	0.70	0.91	0.80	0.91	0.90	0.98
T3	0.78	0.94	0.86	0.94	0.98	0.99
T4	0.97	0.99	0.99	0.99	0.99	1.0

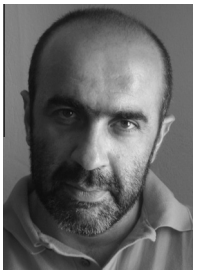
$P_\phi^f = 1 - (1-p)^{\lfloor |Q|_\phi \rfloor}$ and $P_\phi^c = 1 - (1-p)^{\lceil |Q|_\phi \rceil}$ for all traces and p values.

References

[1] T. Beer, M. Fuchs, W. Hoepken, CAIPS: a context-aware information push service in tourism, in: Proc. ENTER'07, 2007, pp. 129–140.
[2] I. Hong, S. Song, J. Par, Location-aware real time positioning with JinifMap, in: Proc. 2nd ICCET'10, 2010, pp. 309–312.
[3] Y. Yang, S. Papadopoulos, D. Papadias, G. Kollios, Spatial outsourcing for location-based services, in: Proc. IEEE 24th ICDE'08, pp. 1082–1091.
[4] Y. Li, J. Han, J. Yang, Clustering moving objects, in: Proc. 10th ACM SIGKDD KDD'04, pp. 617–622.
[5] C.S. Jensen, D. Lin, B. Chin Ooi, Continuous clustering of moving objects, IEEE TKDE 19 (9) (2007) 1161–1174.
[6] J. Chen, C. Lai, X. Meng, Clustering moving objects in spatial networks, in: Proc. 12th DASFAA'07, pp. 611–623.
[7] M.D. Dikaiakos, A. Florides, T. Nadeem, L. Iftode, Location-aware services over vehicular ad-hoc networks using Car-to-Car communication, IEEE JSAC 25 (8) (2007) 1590–1602.
[8] T. Zhu, Y. Zhang, F. Wang, W. Lv, A location-based push service architecture with clustering method, in: Proc. 6th NCM 2010, 2010, pp.107–112.
[9] J. -hong Hong, Z. Su, S. Yeh, A time-driven mobile location aware service for distributing campus information, in: Advances in Geographic Information Science, Springer, 2013, pp. 77–89.
[10] V. Papataxiarhis, V. Riga, V. Nomikos, O. Sekkas, K. Kolomvatsos., V. Tsetsos, P. Papageorgas, S. Vourakis, V. Xouris, S. Hadjiefthymiades, G. Kouroupetroglou, 'MNISIKLIS: indoor location based services for all, in: Proc. 5th LBS and TeleCartography II, LNG& C, 2009, pp. 263–282.
[11] S. Gwynne, E.R. Galea, P.J. Lawrence, L. Filippidis, Modelling occupant interaction with reconditions using the building EXODUS evacuation model, J. Fire Safety 36 (2001) 327–357.
[12] G. Peskir, A. Shiryaev, Optimal Stopping and Free Boundary Problems, ETH Zuerich, Birkhauser, 2006.
[13] Z. Zhang, Y. Yang, A.K.H. Tung, D. Papadias, Continuous k-means monitoring over moving objects, IEEE TKDE 20 (9) (2008) 1205–1216.
[14] C. Anagnostopoulos, K. Kolomvatsos, S. Hadjiefthymiades, Efficient location based services for groups of mobile users, in: Proc. 14th IEEE MDM 2013, 2013.
[15] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Surv. 31 (3) (1999) 264–323.
[16] R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan, Automatic subspace clustering of high dimensional data for datamining applications, in: SIGMOD 1998, pp. 94–105.
[17] D. Fisher, Knowledge acquisition via incremental conceptual clustering, Mach. Learn. 2 (1987) 139–172.
[18] S. Guha, R. Rastogi, K. Shim, CURE: An effcient clustering algorithm for large databases, in: SIGMOD 1998, pp. 73–84.
[19] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice Hall, 1988.
[20] W. Jin, Y. Jiang, W. Qian, A.K.H. Tung, Mining outliers in spatial networks, DASFAA 2006, pp.156–170.
[21] G. Karypis, E.H. Han, V. Kumar, Chameleon: hierarchical clustering using dynamic modeling, IEEE Comput. 32 (8) (1999) 68–75.
[22] R.T. Ng, J. Han, Efficient and effective clustering methods for spatial data mining, in: VLDB 1994, pp. 144–155.
[23] E. Martin, H.P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: SIGKDD 1996, pp. 226–231.
[24] A. Nanopoulos, Y. Theodoridis, Y. Manolopoulos, C2P: Clustering based on closest pairs, in: VLDB 2001, pp. 331–340.
[25] W. Wang, Yang, R. Muntz, STING: a statistical information grid approach to spatial data mining, in: VLDB 1997, pp. 186–195.
[26] M.L. Yiu, N. Mamoulis, Clustering objects on a spatial network, in: SIGMOD 2004, pp. 443–454.
[27] Q. Zhang, X. Lin, Clustering moving objects for spatio-temporal selectivity estimation, in: Proc 15th Australasian Database Conference (ADC), 2004, pp. 123–130.
[28] F. Hppner, F. Klawonn, R. Kruse, T. Runkler, Fuzzy Cluster Analysis, Wiley, 1999.
[29] H.-P. Kriegel, M. Pfeifle, Clustering moving objects via medoid clusterings, in: Proc. 17th SSDBM'2005, 2005, pp. 153–162.
[30] Zhamri Che Ani, Azman Yasin, Mohd Zabidin Husin, Zauridah Abdul Hamid, A method for group formation using genetic algorithm, Intl. J. Comput. Sci. Eng. 2 (9) (2010) 3060–3064.
[31] J.C. Bezdek, S. Boggavarapu, L.O. Hall, A. Bensaid, Genetic algorithm guided clustering, in: Proc. IEEE World Congress on Computational Intelligence, vol. 1, 1994, pp. 34–39.
[32] L.E. Agustn-Blas et al., A new grouping genetic algorithm for clustering problems, Expert Syst. Appl. 39 (2012) 9695–9703. Elsevier.
[33] Yung-Chih Chen, E. Rosensweig, J. Kurose, D. Towsley, Group detection in mobility traces, in: Proc. ACM 6th IWCMC'10, pp. 875–879.
[34] Y. Li, J. Han, J. Yang, Clustering moving objects, in: Proc. 10th ACM SIGKDD KDD'04, pp. 617–622.
[35] T. Zhang, R. Ramakrishnan, M. Livny, BIRCH: an efficient data clustering method for very large databases, SIGMOD Rec. 25 (2) (1996) 103–114.
[36] P. Kalnis, N. Mamoulis, S. Bakiras, On discovering moving clusters in spatio-temporal data, in: Proc. 9th SSTD'05, pp. 364–381.
[37] M. Spiliopoulou, I. Ntoutsi, Y. Theodoridis, R. Schult, MONIC: modeling and monitoring cluster transitions, in: Proc. 12th ACM SIGKDD KDD'06, pp. 706–711.
[38] S. Elnekave, M. Last, O. Maimon, Y. Ben-Shimol, H. Einsiedler, M. Friedman, M. Siebert, Discovering regular groups of mobile objects using incremental clustering, in: Proc. 5th Workshop on Positioning, Navigation and Communication (WPNC 2008), 2008, pp. 197–205.
[39] S. Basagni, Distributed clustering for ad hoc networks, in: Proc. IEEE ISPAN'99, pp. 310–315.
[40] M.J.A. Berry, G. Linoff, Data Mining Techniques for Marketing, Sales and Customer Support, John Wiley & Sons, Inc., 1996.
[41] S. Grossberg, Competitive learning: from interactive activation to adaptive resonance, Cognitive Sci. 11 (1) (1987) 23–63.
[42] T.S. Ferguson, Optimal Stopping and Applications. <http://www.math.ucla.edu/tom/Stopping/Contents.html>.
[43] Y.S. Chow, H. Robbins, D. Siegmund, Great Expectations: The theory of optimal stopping, Houghton Mifflin, Boston, 1971.
[44] L.E. Dubins, H. Teicher, Optimal stopping when the future is discounted, Ann. Math. Statist. 38 (2) (1967) 601–605.
[45] M. Rosenblatt, Remarks on some nonparametric estimates of a density function, Ann. Math. Statist. 27 (3) (1956) 832–837.
[46] A. Zhou, Z. Cai, L. Wei, W. Qian, M-Kernel merging: towards density estimation over data streams, in: Proc. 8th DASFAA 2003, 2003, pp. 285–292.
[47] D.P. Bertsekas, Dynamic Programming and Optimal Control, vol. I, 3rd ed., , 2005, ISBN: 1-886529-26-4.
[48] G. Haggstrom, Optimal sequential procedures when more than one stop is required, Ann. Math. Statist. 38 (1967) 1618–1626.
[49] Y.S. Chow, H. Robbins, D. Siegmund, The Theory of Optimal Stopping, Dover Publications, 1991.
[50] Y.S. Chow, H. Robbins, A martingale system theorem and applications, in: Proc. 4th Berkeley Sympos. Math. Statist. vol. 1, 1961, pp. 93–104.

[51] P.E. Protter, Stochastic Integration and Differential Equations, second ed., Springer, 2003.
[52] X. Hong, M. Gerla, G. Pei, C-C. Chiang, A group mobility model for ad hoc wireless networks, in: Proc. ACM/IEEE MSWiM'99, pp. 53–60.
[53] S.M. Mousavi, H.R. Rabiee, M. Moshref, A. Dabirmoghaddam, MobiSim: a framework for simulation of mobility models in mobile ad-hoc networks, in: Proc. 3rd IEEE WiMob 2007, p. 82.
[54] C. Bettstetter, G. Resta, P. Santi, The node distribution of the random waypoint mobility model for wireless ad hoc networks, IEEE TMC 2 (3) (2003) 257–269.
[55] M.F. Shlesinger, J. Klafter, Y.M. Wong, Random walks with infinite spatial and temporal moments, J. Stat. Phys. 27 (1982) 499–512.
[56] I. Rhee, M. Shin, S. Hong, K. Lee, S.J. Kim, S. Chong, On the levy-walk nature of human mobility, IEEE/ACM TON 19 (3) (2011) 630–643.
[57] N. Aschenbruck, R. Ernst, P. Martini, Indoor mobility modelling, in: Proc. IEEE GLOBECOM 2010 Workshops, pp. 1264–1269.
[58] D. Arthur, S. Vassilvitskii, How slow is the k-means method?, in: Proc. ACM 20th SCG'06, 2006, pp. 144–153.
[59] R. Bellman, Dynamic Programming, Princeton University Press, 1957.
[60] C.J. Stone, Window selection rule for kernel density estimates, Ann. Statist. 12 (4) (1984) 1285–1297.
[61] J.F. Kolen, T. Hutcheson, Reducing the time complexity of the fuzzy c-means algorithm, IEEE Trans. Fuz Sys. 10 (2) (2002) 263–267.
[62] J. Kim, W. Han, W. Choi, Y. Hwang, T. Kim, J. Jang, J. Um, J. Lim, Performance analysis on mobility of ad-hoc network for inter-vehicle communication, Comput. Inf. Sci. (2005).
[63] H. Ehsan, Z.A. Uzmi, Performance comparison of ad hoc wireless network routing protocols, in: IEEE INMIC 2004.
[64] D. Wingate, S. Singh, Kernel predictive linear gaussian models for nonlinear stochastic dynamical systems, in: International Conference on Machine Learning (ICML), 2006, pp. 1017–1024.

Christos Anagnostopoulos received his B.Sc. (2002), M.Sc. (2003), and Ph.D. (2008) in Computer Science from the Department of Informatics and Telecommunications of the National and Kapodistrian University of Athens (UoA). During the period 2009-2010, Christos served at the University of Thessaly, Department of Computer Science and Biomedical Informatics as a visiting Assistant Professor. Since the beginning of 2011, he belongs to the faculty of the Ionian University, Department of Informatics, where he is an Assistant Professor in Network-centric Information Systems. Currently, Christos is also a post-doctoral Research Fellow at the School of Computing Science, University of Glasgow (UoG) and a member of the IDEAS research group at UoG. Since 2004, he has been a member of the Communication Networks Laboratory of the UoA and the Pervasive Computing Research Group (P-COMP) at UoA. Christos has participated and is a co-investigator in numerous projects realised in the context of EU Programs (RAWFIE, IDIRA, IPAC, SCIER, Grabel, PeerAssist, E2R) as well as National Initiatives (Maribrain, Polysema). His research interest is focused on distributed, mobile computing systems and machine learning techniques for Big Data processing. He is the author of over 80 publications, in international scientific journals, conferences and books, in the above areas.



Stathes Hadjiethymiades received his B.Sc., M.Sc. and Ph.D. in Informatics and Telecommunications from the Department of Informatics & Telecommunications (DIT) of the University of Athens (UoA), Athens, Greece. He also received a joint engineering-economics M.Sc. degree from the National Technical University of Athens. In 1992 he joined the Greek consulting firm Advanced Services Group, Ltd., as an analyst/developer of telematic applications and systems. In 1995 he became a member of the Communication Networks Laboratory of UoA. During the period 2001–2002, he served as a visiting assistant professor at the University of Aegean, Department of Information and Communication Systems Engineering. In 2002 he joined the faculty of the Hellenic Open University (Department of Informatics), Patras, Greece, as an assistant professor. Since the beginning of 2004, he belongs to the faculty of UoA, DIT where he presently is an associate professor. He has participated in numerous projects realized in the context of EU programmes and national initiatives. His research interests are in the areas of mobile, pervasive computing, web systems engineering, and networked multimedia applications. He is the author of over 150 publications in these areas.

Kostas Kolomvatsos received the B.Sc. degree in informatics from the Department of Informatics, Athens University of Economics and Business, Athens, Greece, in 1995, and the M.Sc. and Ph.D. degrees in computer science from the Department of Informatics and Telecommunications (DIT), University of Athens (UoA), Athens, Greece, in 2005 and 2013, respectively. He is currently a Researcher in the DIT, UoA and an adjunct professor at the Department of Computer Science, University of Thessaly, Greece. He has authored over 20 publications in his current research areas of interest-mobile and pervasive computing.