Mobile Sinks for Information Retrieval from Cluster-Based WSN Islands

Grammati Pantziou[1], Aristides Mpitziopoulos[2], Damianos Gavalas[2], Charalampos Konstantopoulos[3], and Basilis Mamalis[1]

[1] Department of Informatics, Technological Educational Institution of Athens, Athens, Greece
(pantziou,vmamalis@teiath.gr),

[2] Department of Cultural Informatics, University of the Aegean Mytilene, Lesvos, Greece
{crmaris, dgavalas@aegean.gr}

[3] Department of Informatics, University of Piraeus Piraeus, Greece
konstant@unipi.gr

Abstract. Mobile sinks (MS) mounted upon urban vehicles with fixed trajectories (e.g. buses) provide the ideal infrastructure to effectively retrieve sensory data from isolated Wireless Sensor Network (WSN) fields. Existing approaches involve either single-hop transfer of data from sensors that lie within the MS's range or heavy involvement of network periphery nodes in data retrieval, processing, buffering and delivering tasks. These nodes run the risk of rapid energy exhaustion resulting in loss of network connectivity. Our proposed protocol aims at minimizing the overall network overhead and energy expenditure associated with the multi-hop data retrieval process while also ensuring balanced energy consumption among network nodes and prolonged network lifetime. This is achieved through building cluster structures consisted of member nodes that route their measured data to their assigned cluster head (CH). CHs perform data filtering upon raw data exploiting potential spatial-temporal data redundancy and forward the filtered information to appropriate end nodes.

1 Introduction

A main reason of energy spending in energy-constrained Wireless Sensor Network (WSN) environments relates with transfers of sensor readings, in raw or processed form, from the sensors to remote sinks. These readings are typically relayed using ad hoc multihop routes in the sensor network. A side-effect of this approach is that the Sensor Nodes (SNs) located close to the sink are heavily used to relay data from all network SNs [4]; hence, their energy is consumed faster, leading to a non-uniform depletion of energy in the network [12]. This results in network disconnections and limited network lifetime. Network lifetime can be extended by reducing data relaying energy spending.

Recent research work in the field WSNs has proved the applicability of mobile elements (submarines, cars, buses, mobile robots, etc) for the retrieval of sensory data from SNs in comparison with multihop transfers to a centralized element. A Mobile Sink (MS) moving through the network deployment region can collect data from the static SNs over a single hop radio link when approaching within the radio range of the static SNs or with limited hop transfers if static SNs are located further. This naturally avoids long-hop relaying and reduces the energy overhead at SNs near the base station, prolonging the network lifetime [11].

Several WSN applications involve urban areas that need to be monitored with respect to environmental parameters, surveillance, fire detection, etc. In these environments, individual areas are typically covered by isolated 'sensor islands' wherein a number of SNs located in the periphery of the sensor field can be used as 'rendezvous' points so as to collect sensory data from neighbor SNs and deliver them to a MS when the latter approaches within radio range [13].

In this context, the specification of the appropriate number and locations of Rendezvous Nodes (RN) is crucial. The number of RNs should be proportional to the deployment density of SNs. If a small number of RNs is selected, the energy supplies of those SNs and their neighbors will be rapidly depleted. If, on the other hand, a large number of SNs are appointed as RNs, those nodes will attempt to deliver their collected data simultaneously and they will experience a high number of packet collisions and outages [4], which results in buffer overflows.

Herein, we investigate the use of MSs for efficient data collection from urban 'sensor islands'. We argue that the ideal carriers of such MSs are public surface transportation vehicles that repeatedly follow a predefined trajectory with a periodic schedule that may pass along the perimeter of the isolated sensor fields.

Our proposed protocol called MobiCluster aims at minimizing the overall network overhead and energy expenditure associated with the data retrieval process while also ensuring balanced energy consumption among SNs and prolonged network lifetime. This is achieved through building cluster structures consisted of member SNs that route their measured data to their assigned Cluster Head (CH). The CHs perform filtering upon raw data exploiting potential spatial-temporal data redundancy and forward the filtered information to their assigned RNs, typically located in proximity to the MS's trajectory. We also introduce a method for enrolling appropriate SNs as RNs taking into account the deployment pattern and density of SNs.

The remainder of this article is organized as follows: Section 2 reviews related work in the field. Section 3 presents the execution phases of MobiCluster. Section 4 discusses simulation results that compare the performance of MobiCluster against alternative approaches and Section 5 concludes our work.

2 Related Work

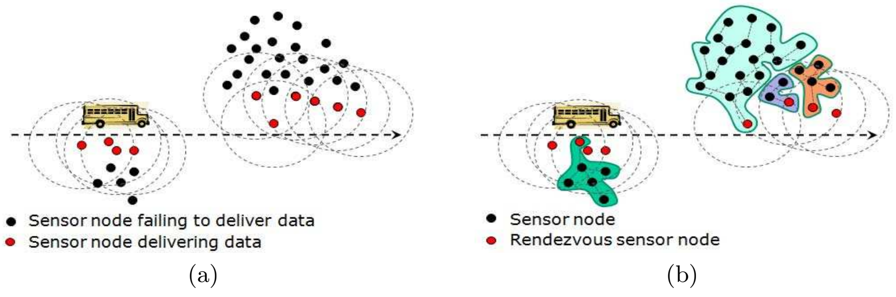
Maintaining connectivity and maximizing the network lifetime stand out as critical considerations in WSNs design. Mobile devices can be used as efficient means for addressing these issues. In urban environments, mobile platforms are already

[4] The term 'outage' is defined as the fraction of SNs which fail to send their data while remaining within the receiver's transmission range [3].

available in the deployment area, e.g. public buses. With sinks mounted upon mobile platforms, the connectivity problem is tackled using MSs that retrieve information from isolated parts of WSNs. Energy efficiency is ensured by MSs traversing or travelling around a WSN field that moderate the energy consumption of SNs by reducing multihop communication.

Existing approaches exploiting sink mobility for data collection in WSNs mainly differ on the properties of sink mobility as well as the wireless data transfer methods [5]. Several approaches target sparse WSNs deployments that suffer from connectivity problems, wherein an MS visits individually SNs and downloads sensory data over a single-hop wireless transmission; the mobility of MSs can be random [1], predictable (their movement pattern is known beforehand) [3] or controlled (their movement is actively controlled in real time) [5, 12]. Rendezvous-based solutions [13] target isolated WSN partitions wherein data are accumulated at designated sensors; these SNs (RNs) buffer collected data until they are relayed to an MS.

Our research targets applications that involve monitoring of isolated urban areas with respect to environmental parameters, surveillance, fire detection, etc. We assume those areas are densely covered by SNs therefore comprising separate urban 'sensor islands'. In such environments, MSs mounted upon city buses that repeatedly follow a predefined trajectory comprise adequate infrastructure for sensory data collection since such vehicles are highly likely to approach the perimeter of the isolated islands.



● Sensor node failing to deliver data
● Sensor node delivering data
(a)
● Sensor node
● Rendezvous sensor node
(b)

Fig. 1. (a) SNs delivering or failing to deliver data in [3]; (b) rendezvous SNs and directed diffusion paths in [12].

In this context, the works presented in [2] and [12] are mostly relevant to the research described herein as they assume mobile data collectors with fixed itineraries. In [3], a network access point was mounted on a public bus moving with a periodic schedule. It is assumed that the mobile node comes within direct radio range of all static SNs, i.e. only single-hop data transfers are possible and the majority of the SNs may fail to deliver their cached data (see Figure 1(a)).

In [12], mobile robots are used to collect data from groups of SNs. During a training period, all the WSN edge SNs located within the range of mobile robot routes are appointed as RNs and build paths connected them with the remainder of SNs. Those paths are used by remote SNs to forward their sensory data to the edge (rendezvous) SNs. Directed Diffusion (DD) [9] is applied upon raw data as they are forwarded from the source to the edge SNs. Similarly to [3], the mobile robots are exclusively used as data collectors. The movement of mobile robots is controllable, which is impractical in realistic urban traffic conditions. Most importantly, no strategy is used to appoint suitable SNs as RNs (all SNs located in proximity to the robot's trajectory are designated as RNs) while selected RNs are typically associated with uneven numbers of SNs (see Figure 1(b)).

Another disadvantage of [12] relates with the use of DD which creates low latency trees from RNs to source SNs. Data from different sources can be opportunistically aggregated at intermediate SNs along the established paths: whenever similar data happens to meet at a branching node in the tree, copies of similar data are replaced by a single message [10]. Also, DD paths are typically prolonged and span large geographical areas, hence they fail to exploit the redundancy inherent in data collected by neighbor source SNs.

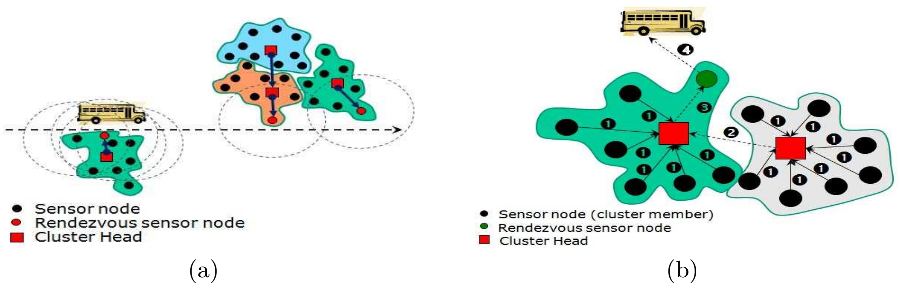


Fig. 2. (a) Rendezvous SNs, cluster structures and data forwarding paths in MobiCluster; (b) sensory data collection and forwarding phases in MobiCluster (numbers indicate the phases time sequence and dotted lines denote inter-cluster traffic and data delivery to the MS).

Our proposed MobiCluster protocol aims at addressing all the aforementioned problems. We propose the use of vehicles not exclusively engaged to data collection (i.e. urban buses) to carry MSs. MobiCluster ensures delivery of data even through multi-hop transfers from source SNs located far from the MS trajectories. Our focus is on building hierarchical cluster structures comprising neighbor SNs to increase the performance of intra-cluster data filtering and minimize the data relaying overhead (see Figure 2). Cluster-based data aggregation is more effective than DD as it is performed upon data derived from SNs located in a restricted geographical space. Emphasis is given on selecting the appropriate RNs among SNs located in the periphery of the sensor islands (so that they remain within the range of MSs for sufficient time and they buffer data from balanced-sized groups of source SNs).

3 Implementation and Execution Phases of Mobicluster

The execution of MobiCluster is divided in five phases described in the following subsections. The first three phases comprise the setup phase while the last two comprise the steady phase.

3.1 Phase 1: Clustering

Clustering has proven to be an effective approach for organizing the network into a connected hierarchy through partitioning SNs into a number of small groups called clusters. Each cluster has a coordinator, referred to as a CH, and a number of member SNs [7]. The member SNs report their data to the respective CHs. The CHs aggregate the data and send them to a remote processing element through other CHs.

To the best of our knowledge, so far clustering has been proposed for efficiently transferring sensory data to static sinks. However, in the particular context of applications wherein MSs monitor isolated urban sensor islands, clustering also exhibits several advantages:

- For 1-hop clusters, cluster members are located maximum 2-hops away, so high redundancy is likely to exist. Hence, raw sensory data may be effectively filtered by CHs, i.e. the energy-expensive processing of raw data is not performed by RNs.
- The majority of network packet transmissions take place between cluster members and CHs. This results in localization of traffic since data traffic is restricted within clusters and not directly routed to RNs.
- Cluster structures imply a more flexible and scalable network organization: in clustered organizations topology changes are dealt with locally, without affecting the whole network.

The LEACH protocol [7] has been among the first proposals in WSN clustering research. LEACH assigns a fixed probability to every node so as to elect itself as a CH. At the end of the clustering process each node decides whether to become a CH or not. SNs take turns in carrying the role of a CH. In [4] unequal clustering has been proposed.

Our clustering algorithm borrows ideas from the above-mentioned approaches and has been designed based on the following principles: (a) cluster formation is a completely distributed procedure; (b) cluster structures are formed within a single iteration; (c) CHs are reachable in a single hop from their cluster members; (d) since CHs are engaged to data processing tasks and also relaying inter-cluster traffic to RNs, they are elected on the basis of their residual energy supply; (e) since CHs closer to the MS's trajectory are burdened with heavier relay traffic and tend to die faster, an unequal clustering approach is followed which groups the SNs into clusters of unequal size i.e., clusters close to the MS's trajectory include less SNs than the other clusters.

The clustering algorithm is detailed as follows. During an initialization phase, the MS moves along its fixed trajectory broadcasting periodically a BEACON signal to all SNs at a fixed power level. Each sensor can compute the approximate distance to the closest location of the MS based on the largest received signal strength. In the sequel, each node of the network with the same probability p becomes a tentative CH. SNs that fail to become tentative CHs remain in sleeping mode until the final CHs are elected. Each tentative CH executes the final CH election algorithm given below (Algorithm 1):

Algorithm 1 Cluster head election

```
1: if $dist(\nu, MS) < d$ then
2:     $\nu.C_{range} = R$
3: else
4:     $\nu.C_{range} = R'$
5: end if
6: broadcast $Competition\_Msg(\nu.Node\_ID, \nu.E_{residual}, \nu.C_{range})$
7: On receiving a $Competition\_MSG$ from a node $u$
8: if $dist(\nu, u) < max(\nu.C_{range}, u.C_{range})$ then
9:     $u$ is added to $N_\nu$
10: end if
11: while the "tentative CH competition time" has not expired do
12:     if $\forall\ u \exists N_\nu, \nu.E_{residual} > u.E_{residual}$ then
13:         broadcast $Final\_CH\_Msg(\nu.Node\_ID)$
14:         exit
15:     end if
16:     if a $Final\_CH\_Msg(u.NODE\_ID)$ is received and $u \exists N_\nu$ then
17:         broadcast $Final\_CH\_Msg(\nu.Node\_ID)$
18:         exit
19:     end if
20:     if a $Quit\_Competition\_Msg(u.NODE\_ID)$ is received and $u \exists N_\nu$ then
21:         $u$ is removed from $N_\nu$
22:         exit
23:     end if
24: end while
```

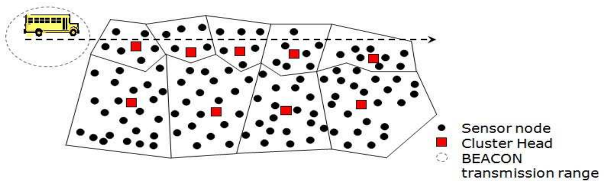


Fig. 3. Unequal cluster formation in MobiCluster.

Each tentative CH decides about the value of its competition range (C_{range}). Two different competition ranges R and R' are allowed. These ranges are used to finally create clusters of two different sizes (see Figure 3). No final CH is allowed within the competition range of another final CH. Each tentative CH elects its competition range based on its distance from the MS (lines 1-4 of Algorithm 1). Namely, if the distance of a tentative CH ν from the MS ($dist(\nu, MS)$), is smaller than a predefined distance d, then ν sets its competition range equal to

R. Otherwise, it sets its competition range equal to $R' = cR$, where c is a small constant, greater than 1.

Once the tentative CHs have decided about their competition ranges, the approach of Chen et al. [4] is employed for choosing the final CHs. First, each tentative CH ν sends a $Competition\_Msg(\nu.Node\_ID, \nu.E_{residual}, \nu.C_{range})$ announcing its residual energy ($\nu.E_{residual}$) and its competition range ($\nu.C_{range}$). Assuming that the broadcast radius of every control message is R', each tentative CH ν constructs the set N_ν of its "competing neighbors" defined as follows:

$$N_\nu = \{\text{tentative CH } u|\ dist(\nu, u)\ <\ max(\nu.C_{range}, u.C_{range})\}$$

i.e., N_ν contains tentative CHs u such that either u is within the range of ν or ν is within the range of u (lines 7-9 of Algorithm 1). If for each node u that belongs to N_ν, the residual energy of u is smaller than the residual energy of ν then node ν sends a $Final\_CH\_Msg(\nu.Node\_ID)$ message announcing its decision to become a final CH to its "competing neighbors" (lines 12-14). Ties can be solved by choosing the smallest ID SNs [4]. If a tentative CH ν receives from a "competing neighbor" u a $Final\_CH\_Msg(u.Node\_ID)$ message it quits competition by sending a $Quit\_Competition\_Msg(\nu.Node\_ID)$ message (lines 16-18). If a tentative CH ν receives from a "competing neighbor" u a $Quit\_Competition\_Msg(u.Node\_ID)$ message, it removes u from the set N_ν.

Once the final CHs have been elected, sleeping SNs wake up and each CH broadcasts a message to announce its election. Each ordinary (non-CH) node uses the received signal strengths to join to the closest CH.

When the cluster formation finalizes, sensory data collected at CHs from their attached cluster members are forwarded towards the RNs following an inter-cluster overlay graph (see Figure 2(b)). The selected transmission range among CHs may vary to ensure a certain degree of connectivity and to control interference [3].

3.2 Phase 2: RNs selection

RNs guarantee connectivity of sensor islands with MSs, hence their selection largely determines network lifetime. RNs are selected among candidate SNs typically located in the periphery of the sensor island and lie within the range of travelling sinks. Suitable RNs are those that remain within the MS's range for relatively long time, in relatively short distance from the sink's trajectory and have sufficient energy supplies. In practical deployments, the number of designated RNs introduces an interesting trade-off. A large number of RNs implies that the latter will compete for the wireless channel contention as soon as the mobile robot appears in range, thereby resulting in low data throughput and frequent outages. A small number of RNs implies that each RN is associated with a large group of sensors. Hence RNs will be heavily used during data relays, their energy will be consumed fast and they are likely to experience buffer overflows.

To regulate the number of RNs and prevent either their rapid energy depletion or potential data losses we propose a simple selection model whereby a set of cluster members (in vicinity to the MS's trajectory) from each cluster is enrolled as RNs. RN role may be switched among cluster members when the energy level of a node currently serving as RN drops below a pre-specified threshold. In addition to lying in a short distance from MS trajectories, the best candidates RNs are the SNs with sufficient residual energy that receive a relatively high number of BEACON packets (i.e. they remain long within the sink's range).

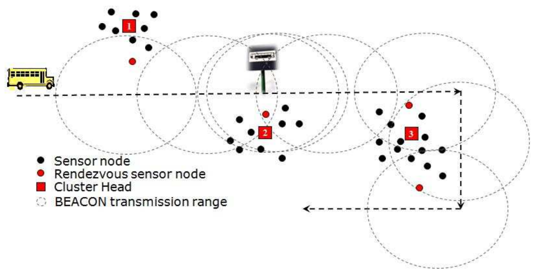


Fig. 4. Selection of RNs in MobiCluster; dashed lines indicate the range of transmitted BEACON packets.

In the topology illustrated in Figure 4 for instance, CH #1 selects as RN the one located closest to the MS's trajectory. CH #2 though appoints as RN the one that received the maximum number of BEACON packets (four beacons) as it resides in short distance from a bus stop (we assume that all SNs share the same energy level). In both cases the appointed RNs remain longer within the MS's range than their neighbor SNs. The cluster led by CH #3 represents a case wherein two cluster members (out of mutual transmission range) are designated as RNs. Such clusters are characterized by an expanded front alongside the MS trajectory and MobiCluster protocol promotes the engagement of multiple SNs as RNs so as to share the data buffering and delivery load.

To count the number of received BEACON packets, when a sensor node receives the i^{th} BEACON, it increases a BEACON counter n_b by one, records the receipt time t_i, the signal strength s_i and restarts a 'Connection Dropped Timer' set equal to $3 \cdot T_{beacon}$ (which allows up to two BEACON packets lost due to channel error). The sensor node also keeps record of the receipt time for the first and last received BEACON, T_{first} and T_{last}.

If a BEACON is received at time $t_{i+1} \approx t_i + n \cdot T_{beacon}$ $(n > 1)$ the SN assumes that $n-1$ BEACON packets have been lost due to channel error or MAC collision and increases n_b by $n-1$. When the 'Connection Dropped Timer' expires the SN assumes that the MS has moved away and the BEACON counter value is finalized.

Then the SN calculates a competence value c_i based on its residual energy, the n_b value and the average signal strength of received BEACON messages (the latter reflects the average distance of the SN from the MS's trajectory). Later on, the SN announces its candidacy to be elected as RN sending to its assigned

CH a *RN_Cand* message containing its $node\_id, ci$, T_{first} and T_{last}. SNs with relatively high c_i values are likely to be elected as RNs. The algorithm executed by SNs receiving BEACON packets is shown below (Algorithm 2).

Algorithm 2 SNs announcing candidacy for RN

```
1: initialize $nb = 0, T_{first} = 0, T_{last} = 0$
2: Wait until a BEACON is received
3: record BEACON receipt time $t_1$ and signal strength $s_1$
4: $T_{first} = t_1, T_{last} = t_i, n_b = 1$
5: start 'Connection Dropped Timer'
6: while 'Connection Dropped Timer' has not expired do
7:     wait until next BEACON is received or 'Connection Dropped Timer' is expired
8:     if a BEACON $i$ is received then
9:         record BEACON receipt time $t_i$ and signal strength $s_i$
10:        $n_b = n_b + \lceil \frac{t_i - t_{i-1}}{T_{beacon}} \rceil$
11:        $T_{last} = t_i$
12:        reset 'Connection Dropped Timer'
13:    end if
14: end while
15: compute $C_i = a_1 \cdot \frac{E_{residual}}{E_{max}} + a_2 \cdot n_b + a_3 \cdot \frac{\sum_{i=1}^{n_b} S_i}{n_b}$
16: broadcast $RN\_Cand(node\_id, c_i, T_{first}, T_{last})$
```

When a CH receives the first *RN_Cand* message it starts a 'RN Candidacy Timer' set equal to the expected MS itinerary time (i.e. the average time required for a bus to arrive from its departure to its end stop). When this timer expires (the sink has moved away) the CH sorts RN candidates in c_i decreasing order list and excludes those with c_i value below a specified threshold T.

Then, it iterates through the candidates list and for each candidate i it examines whether there exist other candidates j where (i, j) are out of mutual transmission range (this is assumed to be true when $T_{j,first} > T_{i,last}$). In case of multiple alternative RN sets, the CH selects as RNs the SNs of a list based on the following priority criteria: a) select the largest node set; b) in case of a tie (two sets with equal number of SNs) select the set with maximum average c_i value.

Using this simple method, it is guaranteed that RN nodes located within the same cluster will not compete each other in the data delivery phase as each will start delivering its data after the previous ends. Hence the wireless channel is more efficiently used, the number of packet collisions is reduced and data throughput is maximized. In addition the employment of multiple RNs, wherever possible, implies lower demand for data buffering space and fair distribution of the energy expenditure associated with data delivery.

3.3 Phase 3: CHs attachment to RNs

An important condition for building inter-cluster overlay graphs is that CHs located far from the MS trajectories attach themselves to a RN node so as to address their clusters' data to them. CHs that include a RN as a cluster member advertise that through broadcasting a *RN_Attach* message to their neighbour CHs. The *RN_Attach* message includes the RN's CH id and a hops counter (initially set equal to 1). Upon receipt of a *RN_Attach* message, a CH increases the hops counter by one and forwards it to its neighbours. Duplicate *RN_Attach* messages (packets with identical CH id value) are dropped. CHs receiving multiple *RN_Attach* messages attach themselves to the RN located minimum hops away to ensure that the inter-cluster transfer of their collected cluster data will incur minimum overhead.

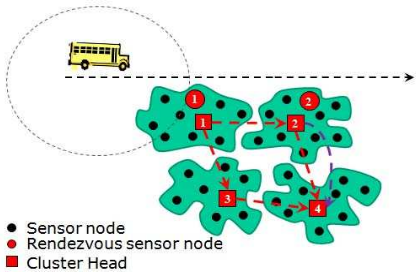


Fig. 5. CHs attachment to RNs.

In Figure 5 for instance, CH #1 advertises its attached RN #1 by sending the message *RN_Attach* $(CH = 1, hops = 1)$. The message is forwarded to all network's CHs (see dotted red lines) with CH #4 receiving *RN_Attach* (1, 2). At this stage all CHs attach themselves to RN #1. Later on, CH #2 designates a RN within its cluster (RN #2) and announces that through broadcasting a *RN_Attach* message (see dotted blue line). Hence, CH #4 will receive a *RN_Attach* (2, 1) message and will choose to attach to RN #2 located only one hop away.

3.4 Phase 4: Data aggregation and forwarding to the RNs

The steady phase of MobiCluster protocol starts with the periodic recording of environmental data from SNs with a T_r period. The data accumulated at individual source SNs are sent to local CHs (intra-cluster communication) with a T_c period (typically T_c is a multiple of T_r). CHs perform data processing to remove data redundancy which is likely to exist since cluster members are located maximum 2 hops away [6]. CHs then forward filtered data towards the cluster where their attached RN belongs to. Alongside the inter-cluster path, a second-level of data filtering may apply.

Upon reaching the end CH, filtered data is forwarded to its local RNs. In the case that multiple RNs exist in that cluster, data are not equally distributed among RNs. Instead, the CH favours the data delivery by the most suitable

RNs, i.e. those with highest competence (c_i) value (see subsection 3.2). Data distribution among RNs should ensure that each RN will be able to accommodate its assigned data, that is deliver all its buffered data and not experience an outage. Hence, the CH sorts its RN list in c_i decreasing order and delivers to each RN node RN_i the maximum amount of data d_i it can accommodate, minus an 'outage prevention allowance' amount O. The d_i value is calculated taking into account the RN's data transfer rate r_i and the time interval t_i that RN_i remains within the MS's range. The process is repeated for each RN_i until all data available at the CH are distributed among the RNs.

3.5 Phase 5: Communication between RNs and mobile sinks

The last phase of MobiCluster protocol involves the delivery of data buffered to RNs. The communication should start when the connection is available and stop when the connection no longer exists, so that the RN does not continue to transmit data when the MS is no longer receiving it. To address this issue we use an acknowledgment-based protocol between RNs and MSs. The MS, in all subsequent path traversals after the setup phase, periodically broadcasts a POLL packet, announcing its presence and soliciting data as it proceeds along the path. The POLL is transmitted at fixed intervals T_{poll}. This POLL packet is used by RNs to detect when the MS is within range. The RN receiving the POLL starts transmitting data to the MS. The MS acknowledges received data packets to the RN so that the RN realizes that the connection is active and the data was reliably delivered. Once the RN transmits a packet to the MS, it starts two timers:

1. Retransmit Timer: The unacknowledged data packet is retransmitted when this timer expires. These retransmissions overcome the effect of packet losses due to channel errors and MAC layer collisions.
2. Connection Dropped Timer: It has similar function with the 'Connection Dropped Timer' used in the setup phase. The RN ceases packet transfers to the MS if it does not receive a POLL message broadcast from the MS before this timer expires.

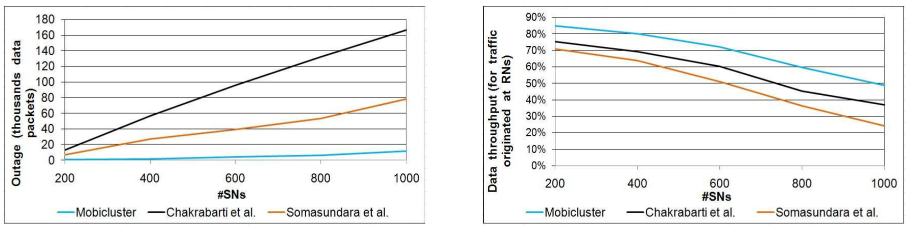
The enrollment of specific SNs as RNs is subject to change during the steady phase. Thus, if the energy supply of a RN falls below a threshold, it may request the local CH to engage another node as RN so as to further extend the network's lifetime. To enable RNs substitution, the CHs polls the RN candidate SNs of the setup phase (excluding the retiring RN) to be informed about their current residual energy status and then selects the new RN candidates list.

4 Simulation Results

MobiCluster has been extensively evaluated with respect to several performance parameters. Its performance has been compared against the algorithms presented in [3] and [12] which are the only existing approaches that involve data collection by mobile observers with fixed itineraries. Unless otherwise specified, the parameters used throughout the simulation tests are those shown in Table 1. The simulation results presented herein have been averaged over ten simulation runs (i.e. ten different network topologies). To ensure a fair comparison between MobiCluster and alternative approaches we assume that MSs follow the same mobility model, wherein the MSs repeat the same trajectory in a periodic basis.

Table 1. Simulation parameters

Parameter	Value
Simulated plane (m^2)	1000 × 1000
#Sensors	100
Sensors transmission power	4 dBm
Sensors transmission range	100 m
Network transfer rate	250 Kbps
Initial sensors battery lifetime (E_{max})	1000 mJ
Energy required for transmission (per byte with 4 dBm transmit power)	50 nJ
Energy required for reception (per byte)	10 nJ
Energy required for data retrieval from sensors (per byte)	2 nJ
Energy required for data fusion (per byte)	5 nJ
BEACON / POLL broadcast period: T_{beacon}, T_{poll}	10 sec
Mobile sink's itinerary repetition period	250 sec
Probability for a node to become tentative CH (p)	10 %
Bytes transmitted from sensors to CHs every time interval	200
Data fusion coefficient (f)	40%



(a) (b)

Fig. 6. (a) Overall number of outages; (b) network throughput

Figure 6(a) illustrates the overall number of outages, i.e. the number of data packets cached in RNs, yet, not delivered to the MS due to buffer overflows, packet collisions or the movement of the MS away of the RNs' transmission range. The algorithm of Chakrabarti et al. [3] performs worse since a large percentage of SNs lies away from the sink's range, hence, they fail to delivery their sensory readings. MobiCluster performs better than [12] because of the more sophisticated selection of RNs which allows them sufficient time to deliver their pre-processed, cashed data. This feature of MobiCluster also justifies its performance gain over alternative methods in terms of network throughput (packets delivered to the MS over those sent from the RNs); those methods employ a large number of RNs without that compete for the contention of the wireless channel contention against other RNs and thus experience a considerable number of packet collisions (see Figure 6(b)). The algorithm of Chakrabarti et al. [3] is shown to perform better than [12], although they involve the same number of RNs (all SNs located within the MS's range), since the latter enables the transmission of sensory data retrieved from all network SNs and thus suffers from higher number of packet collisions.

Figure 7(a) provides an estimate of the total traffic generated throughout the network. MobiCluster outperforms the algorithm of Somasundara et al. mainly due to the improved performance of our cluster-based data fusion method over directed diffusion adopted in [12]. The algorithm presented in [3] incurs lower network overhead as many of the SNs are placed out of the sink's range, thus they do not transmit any packets.

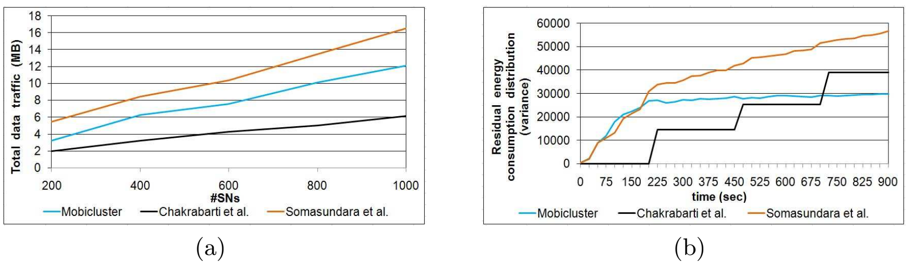


Fig. 7. (a) Total generated network traffic; (b) variance of residual energy consumption

Last, MobiCluster ensures more fair distribution (variance) of energy expenditure among SNs (see Figure 7(b)). This is due to its effective data fusion method and the distribution of the data relay overhead among many CHs due to its unequal clustering organization. Besides, the energy-demanding data processing operation is performed by alternating CHs, while initially selected RNs convey their role to other SNs when their energy level decreases below a specified threshold. The abrupt changes in the graph line corresponding to the algorithm of Chakrabarti et al is due to the simultaneous data transmission of a large part of the WSN SNs towards the MS when the latter appears in range.

5 Conclusions

This paper introduced MobiCluster, a protocol that proposes the use of urban buses to carry MSs that retrieve information from isolated parts of WSNs. MobiCluster mainly aims at maximizing connectivity, data throughput and enabling balanced energy expenditure among SNs.

The connectivity objective is addressed by employing MSs to collect data from isolated urban sensor islands and also through prolonging the lifetime of selected peripheral RNs which lie within the range of passing MSs and used to cache and deliver sensory data derived from remote source SNs. Increased data throughput is ensured by regulating the number of RNs for allowing sufficient time to deliver their buffered data and preventing data losses. MobiCluster moves the processing and data transmission burden away from the vital periphery SNs (RNs) and enables balanced energy consumption among SNs through building cluster structures that exploit the high redundancy of data collected from neighbour SNs.

References

1. Anastasi, G., Conti, M., Di Francesco. M.: Data Collection in Sensor Networks with Data Mules: an Integrated Simulation Analysis. Proceedings of the IEEE Symposium on Computers and Communications (ISCC'2008), 1096–1102 (2008)
2. Blough, D. M., Santi, P.: Investigating Upper Bounds on Network Lifetime Extension for Cell-Based Energy Conservation Techniques in Stationary Ad Hoc Networks. Proceedings of the 8^{th} ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'2002), 183–192 (2002)
3. Chakrabarti, A., Sabharwal, A., Aazhang., B.: Using Predictable Observer Mobility for Power Efficient Design of Sensor Networks. Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks (2003)
4. Chen, G., Li, C., Ye, M., Wu, J.: An Unequal Cluster-Based Routing Protocol in Wireless Sensor Networks. Wireless Networks (2007)
5. Ekici, E.,Gu, Y., Bozdag, D.: Mobility-Based Communication in Wireless Sensor Networks. IEEE Communications Magazine, 44(7), 56–62 (2006)
6. Hall, D.: Mathematical Techniques in Multisensor Data fusion. Artech House, Boston, MA, (1992)
7. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: An Application-Specific Protocol Architecture for Wireless Microsensor Networks. IEEE Transactions on Wireless Communications, 1(4), 660–670 (2002)
8. Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D. E., Pister, K. S. J.: System Architecture Directions for Networked Sensors. Architectural Support for Programming Languages and Operating Systems, 35(11), 93–104 (2000)
9. Intanagonwiwat, C., Govindan, R., Estrin, D., John, H., Silva, F.: Directed Diffusion for Wireless Sensor Networking. IEEE/ACM Transactions on Networking, 11(1), 2–16 (2003)
10. Luo, H., Liu, Y., Das, S.K.: Routing Correlated Data in Wireless Sensor Networks: A Survey. IEEE Network, 21(6), 40–47 (2007)
11. Ma, M., Yang, Y.: Data Gathering in Wireless Sensor Networks with Mobile Collectors, Proceedings of the 22nd International Parallel and Distributed Processing Symposium (IPDPS'2008), 1–9 (2008)
12. Somasundara, A.A., Kansal, A., Jea, D.D., Estrin, D., Srivastava, M. B.: Controllably Mobile Infrastructure for Low Energy Embedded Networks. IEEE Transactions on Mobile Computing, 5(8), 958–973 (2006)
13. Xing, G., Wang, T., Xie, Z., Jia, W.: Rendezvous Planning in Wireless Sensor Networks with Mobile Elements. IEEE Transactions on Mobile Computing, 7(12), 1430–1443 (2008).