Supporting the Sink Mobility: a Case Study for Wireless Sensor Networks

David Tacconi, Iacopo Carreras, Daniele Miorandi
Imrich Chlamtac
CREATE-NET
Via Solteri 38, 38100 Trento, Italy
Email: name.surname@create-net.org

Francesco Chiti, Romano Fantacci
DET Department - Univ. of Florence
Via di Santa Marta 3, 50139 Florence, Italy
Email: name.surname@unifi.it

***Abstract*— This paper deals with a system level design solution to support information gathering in the presence of a mobile querying node that experiences frequent disconnections from a Wireless Sensor Network (WSN), an application scenario that embraces the area of Intelligent Transportation Systems (ITS). The proposed scheme basically relies on the network capability of autonomously adapting to sink mobility in order to properly deliver the requested data. The system is defined as far as its functional elements and related communications protocols, together with validating its effectiveness by means of a simulative study. In particular, the performance in terms of delivering latency and packet delivery ratio has been investigated for several network topologies, architecture and mobility pattern, always highlighting a remarkable quality of provided services together with robustness with regard to operative conditions.**

***Index Terms*— Disconnected Wireless Sensor Networks, Intelligent Transportation Systems, Mobility Prediction, Routing Protocols Design**

I. Introduction

Wireless Sensor Networks (WSNs) [1] have been widely investigated in their various forms over the last decade, but without a clear direction for the introduction of this innovative technology into our daily life. Some of the existing application scenarios embrace monitoring applications [1], [2], where some remote physical phenomenon is constantly controlled by immersing sensors in it and conveying the gathered information to a remote data warehouse. Others focus on event triggered application scenarios [3], such as intrusion or fire detection.

Traditionally, it is assumed a static network architecture composed by sensor nodes deployed over the area to be monitored. Sensor nodes self-organize, and start conveying information, when queried, to a "sink node", which represents the gateway towards the end-user [1]. A major limitation of such approach is that the sink node needs to be directly connected to the remote end-user. Depending from the application scenario and deployment area, this may be extremely inefficient and expensive, since it can potentially require long range communication technologies (i.e., satellite communications, Wi-Fi, Wi-Max, etc.). An alternative solution proposed in the literature consists in letting the information being transported by *data mules* passing by [4], [5], [6]. In this case, the physical movement of nodes (i.e., buses, taxis, cars, pedestrians, etc.) is exploited for providing the communication link between the sensor nodes and the remote end-user. The cost of such strategy is an additional delay to be tolerated on the gathered information, and can be successfully applied to those application scenarios where the timing constraints are not so tight. As an example, we can think at a monitoring application, where a municipal institution is interested in maintaining the historical trends of pollution levels around the city. In this case, buses driving around can collect the readings of the deployed sensors, thus avoiding the need to directly connect the sensors with a backhaul connection.

In this work, we consider a more dynamic scenario, where mobile users passing by query a WSN deployed in the environment. The WSN is thus used for providing real-time information on the surrounding environment [7] (we are not targeting the remote storage of the sensed data). The envisioned application scenario embrace, for instance, the area of Intelligent Transportation Systems (ITS) [8], where the deployed WSN enables cars to obtain the conditions of the surrounding environment while moving, and use this information for eventually take appropriate decisions. As an example, the WSN could be detecting the formation of ice over the road, or monitoring the status of parking slots along the streets of the city center.

Cars or, more in general, mobile users, when needed inject queries into the WSN, which answers, later on, with the requested information, if available. Packets are routed within the network according to an adaptable geographical routing mechanism, where the final destination (sink) is adapted to the mobility pattern of the mobile user querying the network. Hence, differently from a traditional WSN, the *sink* that will finally deliver the requested information is not known a-priori, but depends from the movement pattern of the mobile-user (i.e., speed, direction, uniformity, etc.), and from the network conditions (i.e., load, duty cycle, etc.). Furthermore, the user can in general experience frequent disconnections from the

The work of D. Tacconi, I. Carreras and D. Miorandi has been partially supported by the European Commission within the framework of the BIONETS project IST-FET-SAC-FP6-027748, www.bionets.org.

The work of F. Chiti has been partially supported by the European Commission within the framework of the DustBot project FP6-IST-045299-STREP, *www.dustbot.org*.

WSN, due to a not complete coverage of the region where he is moving. The applied routing mechanism will dynamically adapt to the mobility of the mobile user, and forward packets in order to reach the nearest future position of the sink.
The major contribution of this paper is the definition of a system capable of supporting the described application scenario. The system is defined in all its logical components, and its performance evaluated by means of simulative studies. Results show that the proposed solution is robust to different mobility patterns of the users, and to several network conditions.

The remainder of this work is organized as follows. In Sec. II the most relevant contributions in this area are reviewed. In Sec. III the proposed system architecture is properly introduced, while in Sec. IV the required protocols are described. Sec. V presents the simulation results for evaluating the proposed system performance. Finally, in Sec. VI some conclusions are drawn, pointing out some promising directions for future research in the field.

II. Related Work

In the area of ITS, WSNs are generally adopted for gathering real-time information about road conditions [8], or for accomplishing distributed sensing platforms. In [9], the authors define a WSN for proactively detecting and advertising possible dangerous situations on roads. In [10], a traditional WSN architecture is optimized in order to achieve a high cars detection accuracy, while preserving a sufficiently high network life-time.

When looking at more flexible solutions, a wide attention has been devoted in the last years to the definition of a network architecture together with the related protocols, capable of extending a classical WSN scenario toward the managements of mobile sinks. This might relax the strict constraints in terms of communications and processing resource that typically affect a WSN. In [11], [12] a network architecture where mobile nodes gather information from sensor nodes by means of a single-hop communications is proposed. In this way sensor nodes are not concerned with relaying packets toward remote sink nodes, and with routes maintenance; but the application scenarios are extremely limited. A similar approach is presented in [5], [4], [6], where the nodes mobility is exploited for delivering data from sensors to the sink. As a consequence, it is not needed the sink to be directly connected to a backbone accessible from a remote user and, moreover, the design constraints deriving from multi-hopping packet forwarding paradigm are alleviated. In particular, this strategy results to be particularly efficient in the case of extremely sparse networks.

III. System Architecture

The proposed system architecture is comprised of three kinds of nodes: Mobile Sink (MS), Vice Sink (VS) and Sensor Node (SN). Each type of node is characterized by a different role in the network, and different capabilities in terms of energy, processing, communication and mobility.

A *Mobile Sink* represents the final destination of the information generated by the WSN. MSs are assumed to be moving along a straight road, and to query the Wireless Sensor Network deployed along the same road, when possible. Examples of MSs are buses, regularly moving in the proximity of the WSN, or simply cars, occasionally passing by. MSs run applications, which are augmented with information originating from the WSN, like weather or traffic conditions. An MS is usually equipped with a satellite receiver like GPS, so that information on position, direction and speed is always available. No specific constraints on energy, processing or communication are assumed for MSs. Each MS, while moving, searches for WSNs in the surrounding environment by regularly broadcasting a beacon packet. Upon establishing a mutual communication link with at least one of the WSN nodes, the MS injects a query in the WSN. Each query may include the following information:

- spatial region: the MS is interested in data originating from a limited spatial region, which is defined by the geographical coordinates of its center C_r and its radius R_r;
- MS mobility profile including the current position POS_{MS}, speed V_{MS}, direction DIR_{MS} and timestamp T_{MS}. This information is used by the WSN for setting up the reverse path to answer to the query, as described in the next Section.

Using a SQL-like syntax, Alg. 1 presents an example of a query injected by the MS in the WSN.

Algorithm 1 Example of a query injected in the Wireless Sensor Network by the Mobile Sink.

SELECT temperature
FROM sensors
WHERE region C_r, R_r
MOBILITY POS_{MS} , V_{MS} , DIR_{MS} , T_{MS}

Once the query is accepted by the target WSN, the MS continues moving along the road, while expecting to receive the requested information before exiting the communication range of the overall WSN. In fact, the WSN has a limited amount of time for gathering the requested information and providing it to the MS, since the MS is reachable from the WSN only in proximity of the deployed area.

Vice Sinks are those WSN nodes that are close enough to the MS trajectory (i.e, the road) for communicating directly with the MSs passing by. They represent a sort of *access subnetwork*. These nodes are responsible for initially forwarding the MS query to the other WSN nodes, and for ultimately delivering the requested information to the MS. In addition to this, VSs have sensing capabilities and can be directly queried by the MS. In general, we assume that two consecutive VSs are not connected by a one-hop link, while they are always reachable by the WSN. Being these nodes deployed along the road, no particular energy constraints are assumed over them, since they can be deployed in the proximity of a power supply. Conversely, processing and communication capabilities are limited.

Sensor Nodes are nodes communicating among them by means of multi-hop communications, being a sort of *core network*. They have sensing capabilities, and receive MSs queries from the neighboring nodes. SNs are constrained in their energy, storage and communication capabilities.

The reference scenario is briefly summarized in Fig. 1, where the mobile injects a query on the WSN through the first VS. The query is then forwarded to the interested region (highlighted in grey) where the destination node (the closest to the center of the region) aggregates data of the region of interest by querying other nodes belonging to the same region and then sends collected data toward the target destination. The requested information is first forwarded from the SNs by means of multi-hop communications, and, finally, delivered from the last VS to the mobile sink.

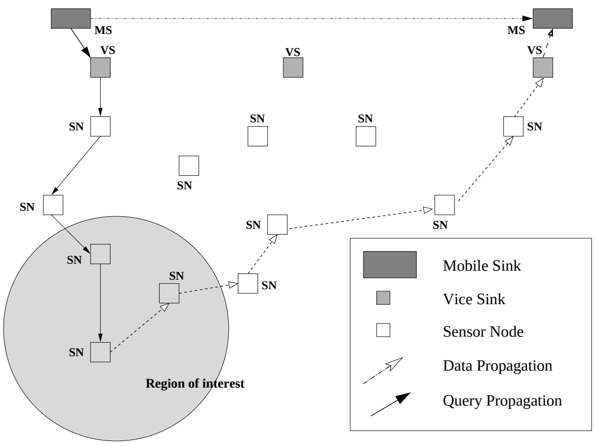

Fig. 1. Example of the considered application scenario. The Mobile Sink injects a query in the WSN through the closest Vice Sink (VS1). The query is forwarded to the queried region (highlighted in grey). The queried data is finally delivered through VS2.

IV. System Protocols

It is assumed that a bootstrap phase is performed in order to initially set up the WSN. During this first phase, each node of the WSN detect its role (VS vs. SN), populate its routing tables with the neighboring nodes and their geographical position[1]. Once this initial phase is over, the WSN is ready to receive and answer MS queries.

A full MS query-response cycle is composed by a *Query Propagation (QP)* phase, where the initially injected query is propagated to the target nodes (i.e., nodes that are located within the region specified in the injected query), and a subsequent *Query Response (QR)* phase, where nodes attempt to forward the queried data toward the most likely MS future position. These phases are described in the following.

A. Query Propagation

Within the communication range of the first VS, the MS sends a *QUERY* message, requesting the information over a limited spatial region. This is achieved by specifying the geographical coordinates of the center and the radius of the region, as described in Alg. 1. The query is then forwarded toward the center of the region of interest by means of geographical routing [13], [14]. Once a node discovers to be the closest one to the center, it stops the geographic forwarding and switches to a restricted flooding of the query limited to the region of interest, electing itself to be the *cluster head* of the region. Each node in the region provides the cluster head with the requested data through the path established by the query flooding. Finally, the cluster head performs a data aggregation algorithm [8] in order to reply to MS's query on behalf of its cluster.

B. Query Response

Once the requested data is aggregated, a *REPLY* message is generated from the cluster head and forwarded toward the estimated future target position of the MS. This is achieved by applying an adaptive routing mechanism, where each node receiving the *REPLY* message estimates the current MS position and decides the next hop accordingly. The adaptive routing strategy can be summarized in the following steps:

1) evaluate the target destination based on the MS mobility information and the actual time;
2) prepare the packet to be forwarded including MS mobility information, data and next hop ID;
3) check whether there is a VS within its neighbors; if so it sends the message toward it, otherwise it selects the closer node to the target destination according to a given distance (for instance Euclidean distance or energy aware distance);
4) if there is no neighbor closer to the target destination this reveals the presence of a *hole*. Under the hypothesis of a high node density, a hole implies that the actual node has neither neighbors closer to the road nor VS to rely data to. Thus it can be applied a *recovering* strategy that routes the packet to the one hop neighbor with the smaller positive angle formed between MS direction and next hop direction, as shown in Fig. 2. This strategy is performed in order to reach the next Vice Sink (according to MS mobility) in the shortest amount of time.

An example of the routing strategy for forwarding *REPLY* message is described in Alg. 2.

At the same time, MS notifies the reached VSs with its novel mobility information. The goal of this strategy is twofold: from one side, MS mobility information are refreshed within the network and whenever a VS receives a message for the MS it is able to revaluate the new position according to new information; from the other side, a VS elected to be the closer

[1]Since we are not interested in an highly accurate estimation of the nodes geographical coordinates, the position of a WSN nodes can be evaluated by applying traditional localization techniques, and assuming to be *a priori* known the VS position, which is along the road.

Algorithm 2 Example of the routing strategy for forwarding REPLY Packet in the Wireless Sensor Network toward the Mobile Sink. The cases of a Sensor Node and of a Vice Sink are considered.

```
if (I AM A SN)
// Sensor Node: evaluate target destination and
// forward packet toward it with the best strategy
   (targetX, targetY) = evaluateDestination(pkt.X_MS, pkt.Y_MS, pkt.speed_MS, t);
   nextHop = findCloserNeighbor(targetX, targetY);
   if (nextHop == NULL)
      holeTurningRouting(targetX, targetY);
   else
      sendPkt(netxHop);
else if (I AM A VS)
// Vice Sink Node: if the MS has already passed by
// re-route the packet to next hop, otherwise wait for MS
   if (receivedMobUpdate == TRUE)
      updateMobilityInfo(Pkt);
      sendPkt(nextHop);
   else
      waitForMS();
end
```

to the target position already knows if the MS has already passed by, and it is able to re-route the packet toward the next VS (according to the direction of the MS) with new mobility information. However, if VS is not aware of the passage of the MS it waits for a fixed time interval before routing the packet again.

Upon reaching the first VS, it can apply one of the following strategies:

1) *Store and Wait Strategy (SWS)*: in this case, if the VS has already received information on the MS, then the *REPLY* packet is iteratively forwarded toward the next VS along the road, until a VS with no information on the MS is found;
2) *Predator Strategy (PS)*: in this case, if the VS has not yet received any information on the MS, the *REPLY* packet is sent toward the previous VS until a VS endowed with information is found. At this point the SWS approach is then applied.

V. Performance Evaluation

In order to evaluate the proposed system performance we resort to extensive numerical simulations with an open source simulation tool [15]. We assumed N_s^2 sensors to be uniformly deployed over a $N \times M$ regular grid, with the distance between two consecutive sensors being 25 m. N_v equally spaced vice-sinks (VSs) are installed along the road. Mobile users are moving with a speed that is uniformly distributed between

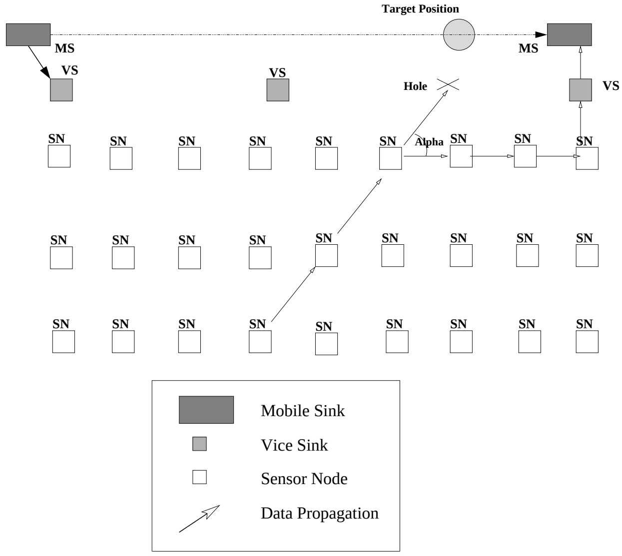

Fig. 2. Routing recovery strategy in the presence of a hole: when a SN tries to reach the target position, but it does not have any node closer and any VS in its neighborhood, it follows MS direction, selecting the neighbor with the closest positive angle alpha between next hop direction and MS target position.

v_{max} and v_{min}. The current speed of the mobile users is updated every 5 s. Nodes communicate at a rate of 250 Kb/s, and adopt CSMA/CA mechanism for collision management.

At first, we have varied the number of sensors by changing the number of raws of sensors M in the grid from 1 to 25, with $N = 40$ and $N_v = 10$. Then we have varied the VSs number V_s with $M = 25$. In both cases also the MS mobility pattern has been varied.

The duty cycled operation has been taken into account by introducing an additional delay to each packet transmission. Sensor nodes have usually a cycle time $C_T = 1$ s and a duty cycle $D_C \in [1\%, 11\%]$. Considering a bit rate of 250 Kb/s, we have accordingly introduced an average delay equal to 150 ms, that corresponds to a duty cycle $D_C = 1.5\%$. This value has been derived by considering the saturation condition in which the bit rate is equal to 150 Kb/s [2] and the packet rate is $P_R = 521 pkt/s$ given a packets length of 36 bytes. Then, over a cycle time, it is possible to transmit up to $P_R \times C_T \times D_C = 6.7$ pkt/s that corresponds to an average maximum delay of 150 ms among consecutive packets. This represents a sort of worst case for the latency affecting the network performance at the MS side.

We first consider a network with $N_v = 10$ VSs equally spaced over a 1000 m long road, with a variable number of sensors in the network. MS injects a query directed toward a randomly selected region reaching the vice sink $Vs\,(i)$, with i uniformly distributed among 0 and $N_v/2$. Once the *REPLY* packet is delivered to MS or MS has reached the end of the road, the simulation is stopped. For each value of M we run

[2] In fact, it is well known that a CSMA/CA data-link layer introduce an overhead of 40%.

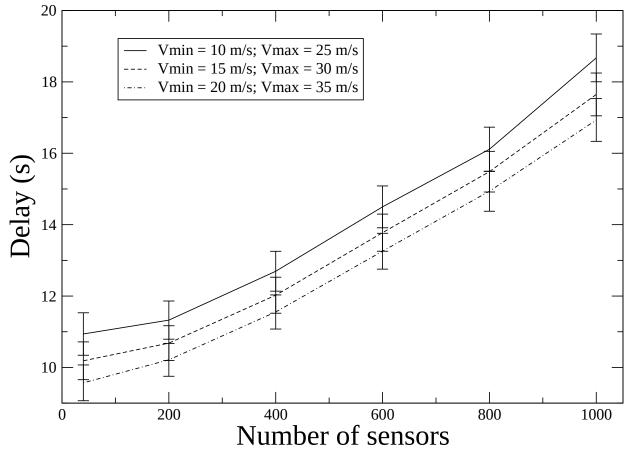

Fig. 3. Latency between query packet injection and reply packet reception at Mobile Sink with 10 VS uniformly distributed over a 1000 m long road, variable number of sensors in the network distributed over a grid, distance among sensors 25 m, MS speed updated every 5 s uniformly distributed within $[10, 25]$ m/s for the first mobility pattern, $[15, 30]$ m/s for the second mobility pattern and $[20, 35]$ m/s for the third mobility pattern.

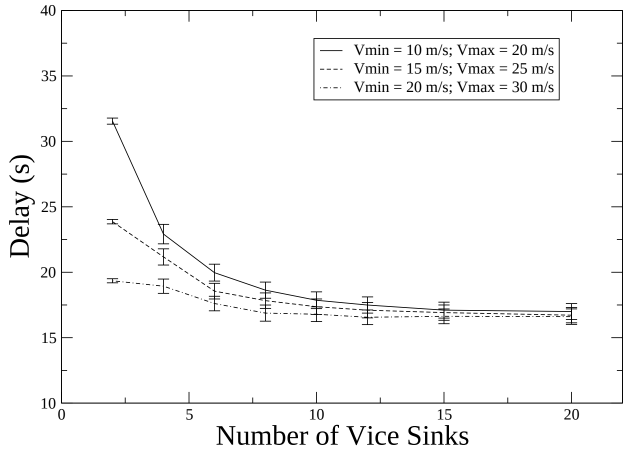

Fig. 4. Latency between query packet injection and reply packet reception at Mobile Sink with variable VSs, 1000 sensors distributed over a grid, distance among sensors 25 m, MS speed updated every 5 s with values uniformly distributed within $[10, 20]$ m/s for the first mobility pattern, $[15, 25]$ m/s for the second mobility pattern and $[20, 30]$ m/s for the third mobility pattern.

	MS Speed: Vmin, Vmax (m/s)		
$N \times M$	$10, 25$	$15, 30$	$20, 35$
40	1.00	1.00	0.978
200	1.00	1.00	0.992
400	1.00	1.00	0.99
600	1.00	0.994	0.974
800	1.00	0.99	0.95
1000	1.00	0.972	0.904

TABLE I

PACKET DELIVERY RATIO, WITH A VARIABLE NUMBER OF SENSORS AND $N_v = 10$, AND A VARIABLE RANGE OF SPEEDS.

	MS Speed: Vmin, Vmax (m/s)		
N_v	$10, 20$	$15, 25$	$20, 30$
2	0.988	0.932	0.912
4	1.00	0.992	0.936
6	1.00	0.992	0.954
8	1.00	0.992	0.964
10	1.00	0.99	0.954
12	1.00	0.99	0.936
15	1.00	0.994	0.964
20	1.00	0.986	0.932

TABLE II

PACKET DELIVERY RATIO, WITH A VARIABLE NUMBER OF VICE SINKS AND $M = 25$, AND A VARIABLE RANGE OF SPEEDS.

500 different simulations and evaluated the average latency for received packets between query injection and reply reception, with a confidence interval of 98 $\%$ and the packet delivery ratio (PDR) defined as the ratio among received packets and injected queries.

Results with three different patterns of mobility are shown in Fig. 3 and Table. I. It can be noticed that for each mobility pattern the delay increases as the size of the network increases; also it is worth verifying that delay increases as the range of speeds decreases but at the same time PDR goes down from 100 $\%$ to 90.4 $\%$ when MS is faster. This result is somehow expected, since when the MS speed spans between higher values, MS is able to cover in a lower amount of time those parts of the road where there is no connectivity with any VS, thus decreasing the latency. At the same time, however, an increase of speed range results in higher probability that the network is not able to deliver the *REPLY* packet to the MS before it looses connectivity with the last VS. These results show that the proposed routing protocol is robust even for a large WSN, and that it is not affected by MS speed increasing as far as the delivering latency is concerned, although the MS experiences a lower PDR, which is due to physical limits of the network.

Finally, a network topology has been simulated, in which the number of raws of sensors is fixed to $M = 25$ (i.e. 1000 sensors in the network) and the number of VSs N_v is varied from a minimum of 2 to a maximum of 20 deployed at a distance of $1000/N_v$ so that every VS is always connected to one sensor node and there is no connectivity between consecutive VSs. The average delay and the PDR have been evaluated running 500 simulations for each value of N_v and considering a confidence interval of 98 $\%$.

Results are shown in Fig. 4 and Table. II where it can be highlighted how increasing the number of VSs does not affect the measured latency. It can also be observed that latency reaches a floor value of 17 s for each mobility pattern with a number of VS higher than 10. This is a relevant result, since one should expect that with a shorter distance among VSs the waiting time and, consequently, the latency would have sensibly decreased.

Nevertheless, we point out that for a given mobility pattern there is a number of VS beyond which the MS does not experience an increase of PDR (since a response failure happens for those queries addressed toward very far regions and the network does not have the time to reach the MS before it disconnects to the last VS) and the latency goes to a floor value. We have simulated three mobility patterns in which the MS does not experience a PDR below $90\ \%$, so that latencies could be measured for a great deal of cases in order to effectively compare results.

VI. Conclusions and Future Work

As Wireless Sensor Networks (WSN) and embedded devices are becoming more and more part of our every day life, the traditional approach of a sink always connected to the end-user in order to gather queried information is becoming a severe constraint for common user applications. Mobile users passing by on a road and querying a WSN deployed in the environment in order to provide real-time information on the surrounding environment seems to be a scenario that embraces several applications of great interest nowadays, like for instance, the area of Intelligent Transportation Systems (ITS).

We proposed an adaptive routing technique able to reach a sink moving along a straight road even if it experiences frequent disconnections from the WSN. Three different types of nodes have been introduced, defining their role in the network based on the position they assume with respect to the road: Mobile Sink (MS) i.e. the sink moving along the road, Vice Sink (VS) i.e. a node able to directly communicate with MS and Sensor Node (SN) i.e. a node that performs query and response forwarding but can reach MS only through a VS. We considered a simplified scenario with SNs regularly deployed in a grid and VSs regularly disposed along a straight road where the MS is moving frequently updating its position. By considering different ranges of speed for the MS and by varying the number of SNs and the number of VSs, we measured performances of our adaptive routing protocol in terms of latency and packet delivery ratio (PDR).

We have shown by means of an extensive simulation study that the proposed protocol is robust in terms of latency and PDR to different mobility pattern while varying the dimension of the network. We have also observed varying the number of Vice Sinks along the road that there is a density of VSs upon which results do not change even increasing the speed range. All the results have been evaluated with an average maximum delay of 150 ms among consecutive packets that represents a sort of worst case for the latency affecting the network performance at the MS side.

In this work we have shown the physical limits that affect a mobile sink gathering data from a deployed WSN, putting the basis of future studies in the field of WSNs for ITS. Future work will be devoted to a deeper investigation of the proposed algorithms, focusing on supporting different mobility patterns for multiple users experiencing disconnections in a more realistic deployment of WSN nodes. Energy efficiency protocol able to support heavier traffic conditions, data aggregation and data fusion algorithms will be also investigated in order to complete the study on the depicted application scenario.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.

[2] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An analysis of a large scale habitat monitoring application," in *In Proc. of SenSys*, New York, NY, USA, 2004, pp. 214–226.

[3] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh, "Monitoring volcanic eruptions with a wireless sensor network," in *Proc. of EWSN*, Istanbul, Turkey, January 2005.

[4] M. M. B. Tariq, M. Ammar, and E. Zegura, "Message ferry route design for sparse ad hoc networks with mobile nodes," in *Proc. of ACM MobiHoc*, Florence, Italy, 22-25 May 2006, pp. 37–48.

[5] R. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: Modeling a three-tier architecutre for sparse sensor networks," in *Proc. of IEEE SNPA*, Anchorage, AK, USA.

[6] W. Wang, V. Srinivasan, and K. Chua, "Using mobile relays to prolong the lifetime of wireless sensor networks," in *Proc. of ACM MobiCom*, Cologne, Germany, 2005.

[7] W. Lindner and S. Madden, "Data management issues in disconnected sensor networks," in *GI Jahrestagung (2)*, 2004, pp. 349–354.

[8] F. Zhao and L. Guibas, *Wireless Sensor Networks: An Information Processing Approach*. Elsevier/Morgan-Kaufmann, 2004.

[9] M. Karpinski, A. Senart, and V. Cahill, "Sensor networks for smart roads," in *PerCom Workshops*, Pisa, Italy, 2006, pp. 306–310.

[10] S. Coleri, S. Y. Cheung, and P. Varaiya, "Sensor networks for monitoring traffic," in *Proc. of Allerton Conf.*, September 2004.

[11] L. Tong, Q. Zhao, and S. Adireddy, "Sensor networks with mobile agents," in *Proc. of IEEE MILCOM*, 2003.

[12] L. Song and D. Hatzinakos, "Architecture of wireless sensor networks with mobile sinks: Sparsely deployed sensors," *submitted to IEEE TVT*, 2006.

[13] Y. Yu, D. Estrin, and R. Govindan, "Geographical and Energy-Aware Routing: A recursive data dissemination protocol for wireless sensor networks," UCLA-CSD, Tech. Rep., 2001.

[14] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for wireless sensor networks," in *Proc. of MobiCom*, 2000.

[15] OMNeT++ discrete event simulation system. [Online]. Available: http://www.omnetpp.org