# Controlled Vs. Uncontrolled Mobility in Wireless Sensor Networks: Some Performance Insights

Stefano Basagni
Department of Electrical and Computer Engineering
Northeastern University
`basagni@ece.neu.edu`

Alessio Carosi and Chiara Petrioli
Dipartimento di Informatica
Università di Roma "La Sapienza"
{`carosi,petrioli`}`@di.uniroma1.it`

*Abstract*— **Among the many ways of improving the performance of a wireless sensor network (WSN) in terms of crucial metrics such as its lifetime and data latency, exploiting the mobility of some of the network components has been recently observed to be among the most promising. In this paper we demonstrate how two very different schemes for WSN mobility leads to different benefits for network performance. More specifically, we consider the data MULEs kind of random, uncontrolled mobility with single-hop data collection and we compare it with the controllable mobility of the data collection point (sink) where sensor-to-sink data routing follows multi-hop paths. Through quite thorough ns2-based simulations we show that data MULEs are to be used in those WSNs deployed for delay tolerant applications. Benefits of this scheme include low energy consumption and easier protocol and nodal design. Data latency, however, can be unbearably high. We therefore show that a good tradeoff between network lifetime gains and data latency increases can be obtained by using those solutions where the mobility of the sink is controlled by the network conditions.**

## I. INTRODUCTION

The paper explores ways for using the mobility of network components in large multi-hop networks of resource constrained devices, like wireless sensor networks (WSNs) [1], [2]. The aim is that of improving the performance of these networks in terms of network lifetime, throughput and connectivity without significantly affecting data routing and end-to-end latency. Whereas energy efficient solutions have been proposed for WSNs with statically placed sensors and one or more static data collection points (the *sinks*), our investigation delves into different ways some mobile relays or the sinks can be made mobile for improving network performance. In this paper we demonstrate the advantages of using mobility from the perspective of *controlled* vs. *uncontrolled* movements of selected network components. As an example of mobility *not controlled* by the network, we investigate the *data MULEs* approach [3] for which a number of collector nodes are sent roaming through the sensor nodes to provide enhanced connectivity and one-hop data collection. The collectors eventually pass by the sinks, to whom they transfer the gathered data. This mobility scheme is compared with recent solutions where communication is multi-hop from/to the sink, and the sink moves to those areas in the network where the sensor nodes have the highest residual energy (controlled mobility) [4], [5].

This paper aims at providing evidence of the advantages of using mobility and more importantly, at giving insights on the challenges and trade-offs one has to face when designing protocols for mobile WSNs. In particular, after describing the data MULEs solution and other sink mobility approaches, we will focus on a performance evaluation of the two different paradigms. We start by discussing the data MULEs approach to mobility investigating the impact of the number of MULEs, the size of the deployment area, the nodal queue size, and the traffic load on the overall system performance. We then proceed by comparing multiple schemes for sink mobility in WSNs with multi-hop data routing, analyzing the resulting trade offs between network lifetime and data latency. A comparison between these results and those obtained for the data MULEs architecture allows us to clearly identify pros and cons of the two paradigms. We observed that data MULEs are key players in those scenarios where energy conservation is of the essence. However, in this case, the price to pay is the significantly high end-to-end data latency. Four order of magnitude higher delays were observed in the realistic WSN scenarios we considered with respect to the case when the sink moves.

The paper is organized as follows. Sections II and III introduce to the data MULEs and to the sink mobility solutions, which we compare in this paper. Simulation-based performance is then shown in Section IV. We conclude the paper with Section V.

## II. DATA MULES

The data MULEs architecture [3], [6] is comprised of three layers. The bottom layer is made up of the sensor nodes, deployed statically and possibly in large number. Each node has wireless capability. In the middle roam the MULEs. These are nodes characterized by mobility, large storage as well as renewable power. They have the ability of communicating with the sensor nodes as well as with the sinks wirelessly. Most importantly, MULEs movements cannot be predicted in advance, and are considered completely random (uncontrolled mobility). The highest layer of the architecture comprises one or more access points, i.e., the sinks. These are the resource-rich, static components to whom the MULEs deliver the data collected throughout the network.

The concept of a *Mobile Ubiquitous LAN Extension* (MULE) stems from the idea of deploying mobile agents for carrying information [7], especially among nodes in possibly

1-4244-0264-6/07/$25.00 ©2007 IEEE

disconnected networks. MULEs are resource-rich nodes that roam freely throughout the network. When as a result of its motion a MULE gets to be in the radio proximity of a sensor, it receives all the packets generated by that sensor so far (if any). Upon getting close to one of the network *access points* (i.e., the sinks), the MULE transfers to it all the collected packets. In this way, whether there is a route from a sensor to one of the sinks or not, i.e., independently of network connectivity, a packet is eventually delivered to the sink. Communication in the MULEs architecture is single-hop: A node stores the sensed data until a MULE passes by. Once the MULE arrives, the node transmits all the stored packets to the MULE. Given the limited storage capacity of a sensor nodes, chances are that if a MULE does not pass by, some of the packets have to be discarded. When the MULE passes by the sink it transmits to it all the packets it has collected.

## III. SOLUTIONS FOR SINK MOBILITY

In this section we review recent solutions for sink mobility in scenarios where communication from/to the sink is multi-hop. In [4], [5] a mathematical model is defined that optimizes network lifetime by moving the sink among a finite given set of sites (the sink sites), and computes the optimal sojourn time at those sites (this scheme is termed OPT). At the same time, a distributed scheme, called GMRE for Greedy Maximum Residual Energy, is also provided which is more suited to the nature of WSNs. The main idea behind the GMRE distributed protocol is that of controlling the movements of a mobile sink toward the zones of the network where nodal residual energy is higher. This leads to a balanced energy consumption throughout the network, and to a longer network lifetime.

The GMRE network architecture is made up of two layers. The higher layer comprises the sink, a mobile device, not constrained by specific communication, computation and storage capabilities, that collects (and possibly elaborates) all the data coming from the network. The lower layer of the GMRE architecture comprises the sensor nodes. These devices, usually constrained in energy and computational resources, are deployed statically. Their task is to sense data and deliver them to the sink via multi-hop routing, unless they are in the radio vicinity of the sink. Periodically the sink greedily selects the next place to move to among the sites that are at most $d_{MAX}$ far from its current one. While at a site, where it stays at least $t_{min}$ time, the sink probes specific nodes (called *sentinels*) which are in the proximity of potential future sites. After $t_{min}$s, based on the information provided by the sentinels that concerns the residual energy of the nodes around the corresponding site, the sink decides whether to move to a new site or to stay at the current one. In particular, it moves if the minimum residual energy around one of the adjacent sites is higher than the energy around the sink current position. If the sink moves, it will flood the network with a corresponding message, thus stopping the nodes from keeping sending the packets (which will be stored). Once at the new site, with a new flood, the sink will inform the sensor nodes that they can start sending the packets toward its new position. This second flood is also functional to build new routes to the sink. (The reader is referred to [5] for further details and discussion on the GMRE protocol.)

Two other schemes are considered in our comparative performance evaluation. One concerns uncontrolled mobility of the sink. Every $t_{min}$s the sink randomly moves to one of the adjacent sites (including the current position). This simple scheme is termed RM, for Random Mobility.

Finally, we implemented the solution proposed in [8] where the sink cyclically moves along the perimeter of the deployment area. We call this solution Perimeter Mobility or PM, for short.

## IV. CONTROLLED VS. UNCONTROLLED MOBILITY: COMPARISON AND EXPERIMENTS

In this section we illustrate the results of a ns2-based comparative performance evaluation for assessing pros and cons of a typical multi-hop WSN architecture (static sensor nodes) with controllable sink mobility vs. the data MULEs solution.

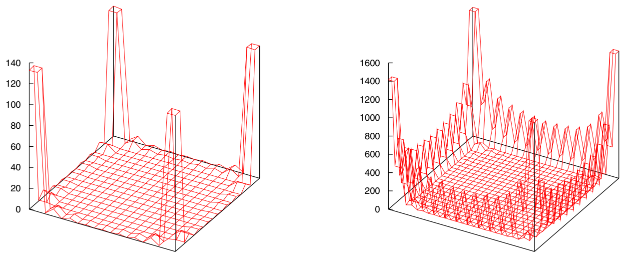
Our evaluation is organized into three parts. First, we delve into the performance of the data MULEs solution, studying the impact of different number of MULEs, nodal buffer size, area size, etc. We then move to sink mobility in scenarios with multi-hop data routing. We compare the performance of OPT, GMRE, PM and RM in different settings, showing the overall performance advantage of controlled sink mobility à la GMRE. The third part concerns comparing the MULEs paradigm and GMRE in realistic and comparable scenarios (i.e., networks with one sink and one MULE). We leverage on the results gained in the first two steps to show that controlled mobility is the solution of choice whenever end-to-end latency is the main concern of the network application. The data MULEs solution is instead more suitable for delay-tolerant, energy-constrained applications.

All our results refer to a basic scenario where nodes are deployed in a grid fashion in a square area. They periodically generate packets at a rate of 0.5bps (low traffic scenario, typical of sensor networking). Packet size is equal to 512B. The buffer size of the sensor nodes varies among 10, 50 and 100 packets. Energy consumption follows the TR1000 specifications [9]. In all our experiments every node generates over 3000 packets during the simulation time. We have run enough experiments to achieve $95\%$ confidence and error within $5\%$.

### *A. Single-hop routing: Data MULEs*

We consider scenarios with 1, 2 and 4 data MULEs. The deployment area is divided into cells. The MULEs move from the center of one cell (source) to the center of another one (destination) according to the random waypoint model [3]. We have selected a squared deployment area, where 361 and 729 sensor nodes are deployed on a grid of side 400m and 572m respectively (with the same node's density). Moreover, in respect to these two scenarios, the MULEs can move on a grid with 81 ($19 \times 19$) and 169 ($27 \times 27$) cells. The MULE speed is set to 1m/s. While going from the source to the destination cell, a MULE stops at intermediate cells for collecting data. When in a cell, the MULE is able to communicate with all (and only) the sensor nodes located in that cell. The time spent in each cell is set to be $\geq$ 1s. MULEs queues are considered unbounded (1000 packets, well above what needed in the considered scenarios). MULEs report to a sink that is (statically) placed at the center of the deployment area. When in proximity of the sink a MULE stops for the time needed to empty its queue.

Fig. 1 displays the number of packets discarded by the sensor nodes in the case of one MULE roaming among sensors with buffer size set to 10 (packets).

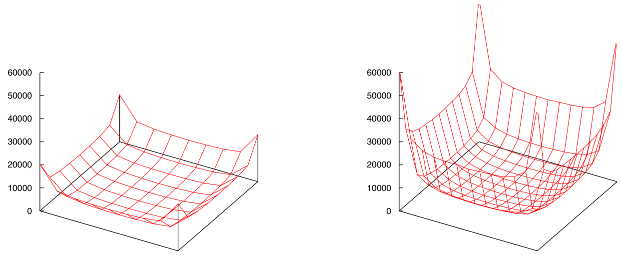


(a) $L = 400$m (b) $L = 572$m

Fig. 1. Average number of discarded packets

Although, on average, there is no noticeable packet loss in the $L = 400$m scenario, we notice that there are a few nodes that, given the presence of a single MULE and small buffers, have to drop some packets. In this case the culprit is the movement of the MULE itself, given that, as it is well-known, the random waypoint-dictated mobility leads the MULE toward the center of the deployment area most of the time. The permanence at the center implies a MULE longer inter-arrival times at the cells on the area borders. This imposes a 4.5% packet loss at the nodes at the four corners, for the scenario with $L = 400$m. In the $L = 572$m scenario, the nodes on the corners drop almost half of the generated packets, due to buffer overflow. Nodes on the perimeter loose almost 20% of their packets. The bad effects of the random waypoint-dictated mobility also result in packet losses in case nodes have a buffer size of 10 packets, and two MULEs roam through the deployment area.

Fig. 2 concerns the inter-arrival times of 1 MULE at the same cell. In all the considered scenarios the inter-arrival time values are extremely high (in the thousands of seconds or more), explaining why the data MULEs approach results in high packet latency. As the number of MULEs increases, it takes less and less time before a MULE arrives at a given cell, as expected. At the same time, given that the MULEs tend more to visit central cells than peripheral ones, the inter-arrival times are higher at the area borders. Inter-arrival times for $L = 572$m are in the order of tens of thousands of seconds: Around three times higher than when $L = 400$m. This is because of the higher number of cells and of the larger distances the MULE has to travel. As we consider low traffic scenarios and a sojourn time of at least 1s, we observed that a MULE tends to stay 1s in every cell it visits (one second is enough to gather the data from the sensors) except for the cell where the sink is located. In the central cell the MULE tends to stay more than 1s, depending on the number of data collected. This time is also inversely proportional to the number of roaming MULEs.



(a) $L = 400$m (b) $L = 572$m

Fig. 2. Average inter-arrival times (s) of 1 MULE

In Table I we show the average latency incurred by data packets from the sensor to the sink for increasing nodal queue size and $L = 572$m, which is the scenario with the highest end-to-end delays.

TABLE I

PACKET LATENCY (SEC), $L = 572$M

| **Nodal queue size** | **1 MULE** | **2 MULEs** | **4 MULEs** |
|---|---|---|---|
| 10 | 21362.39 | 11872.92 | 7245.28 |
| 50 | 23704.73 | 12085.83 | 7254.63 |
| 100 | 23796.53 | 12094.51 | 7443.90 |

We observe that, while it has an impact on the number of lost packets, the size of a node queue has little influence on the packet latency. Packets are transferred from the source queue (node or MULE) to the destination queue (MULE or sink) all together, and the transfer time is quite little. Therefore, the end-to-end delay is given almost entirely by the time from when the packet is generated and a MULE arrives at the cell, plus the time it takes to the MULE to deliver it to the sink. With respect to when $L = 400$m, the latency is almost three times higher. For a small buffer size (10 packets) and a fixed number of data MULEs in the network, the average latency experienced by packets is lower than for the case with bigger buffer sizes (50 or 100 packets). This might appear counter-intuitive, but it is explained by the fact that with smaller buffers packets are lost for overflow. These lost packets are mostly generated by nodes at the border of the deployment area. If they would be collected by a MULE, they would experience a particularly high delay. Since such “critical” packets are not delivered to the sink, they are not considered in computing the average latency, which is, as a consequence, lower.

The end-to-end packet latency *almost* halves with doubling the MULEs, since only one of the delay component halves. The time a packet sits on its node queue depends on the number of MULEs, as expected. When the number of MULEs doubles, this time halves. We observe that this component is the most significant one. The time spent by a packet on a MULE's queue is instead independent of the number of MULEs.

### B. GMRE and multi-hop routing solutions

We consider a scenario in which $361$ sensor nodes are deployed on a $19 \times 19$ grid covering a squared area of side $L = 400$m (this is the same scenario we considered for the data MULEs). Sensor nodes have a transmission radius equal to 30m, which imposes a maximum of 8 neighbors per node. The sink moves among $8 \times 8$ sink sites (grid deployment) at the speed of 1m/s. A shortest path-like routing is used to deliver data in a multi-hop fashion from the sensor to the current site of the sink. The parameter $t_{min}$ is set from 50000s (the best performing value among the many tested) to 1000000s. The parameter $d_{MAX}$ is equal to 190m. We have implemented in ns2 the schemes previous described, namely OPT, GMRE, PM and RM, along with the scenario where the sink is (optimally) placed at the center of the deployment area (called STATIC below).

Fig. 3 shows the average network lifetime achieved by STATIC, RM, PM, GMRE and OPT in the considered scenario. (The network lifetime has been defined as the time until the first node dies because of energy depletion.)

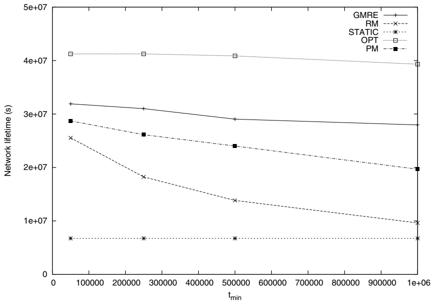


Fig. 3. Average network lifetime for different mobility schemes

TABLE II

IMPROVEMENTS OVER STATIC (%)

| $t_{min}$ | **OPT** | **GMRE** | **PM** | **RM** |
|---|---|---|---|---|
| 50000s | 512 | 374 | 326 | 279 |
| 1000000s | 484 | 315 | 192 | 42 |

Table II shows the improvements on network lifetime with respect to STATIC, which are obtained with all the analyzed heuristics, even for higher values of $t_{min}$. For distributed heuristics such as GMRE, RM and PM we observed that lower $t_{min}$s correspond to higher the network lifetime. For large $t_{min}$s the sink cannot finely decide the time to spend at each site, being stuck at a site for a longer time. This imposes a less uniform energy consumption at the nodes. When $t_{min}$ is very high it is not even possible for the sink to sojourn at all network sites, since the lifetime is reached before the sink can visit them all. Differently from OPT, in the described sink mobility heuristics the decision about whether to move or not is due every $t_{min}$ (after which the sink often moves). This has a detrimental effect on the network lifetime. First of all, because increasing $t_{min}$s have greater effect on GMRE, RM and PM than on OPT in terms of fine tuning the sojourn time at a site. Furthermore, being OPT able to select optimum sojourn times requires much lower sink mobility, which corresponds to lower overhead and energy consumption for route construction and release. GMRE, RM and PM do not have a global view of the network topology and do not know the network traffic, i.e., how the nodes energy consumption evolves over time. OPT takes advantage of this knowledge. The RM and PM heuristics do not enforce any energy-based criterion for sink movement, resulting in the worst performance among all the schemes with mobile sink we investigated. Even if GMRE takes into account nodes residual energy, the decision about whether to move or not and where, is based on a partial knowledge of the energy status of the network. The greedy nature of GMRE choice could lead to a bad move with respect to global network lifetime maximization. The impact of such bad move is clearly higher for high $t_{min}$s since the consequence of a bad move is paid for a longer time.

Table III shows the data latency for $t_{min} = 50000$s and 1000000s (intermediate values are consistent with these ones).

TABLE III

AVERAGE PACKET LATENCY (SEC)

| $t_{min}$ | 50000 | 1000000 |
|---|---|---|
| **OPT** | 0.19 | 0.19 |
| **GMRE** | 0.214 | 0.214 |
| **RM** | 0.175 | 0.18 |
| **PM** | 0.211 | 0.212 |
| **STATIC** | 0.13 | 0.13 |

Despite sink mobility increases the network lifetime sink movements impose higher end-to-end data packet delays. Packets that are generated while the sink is moving and those in transit toward the sink have to wait until routes to the new position of the sink are established before they can resume their route to the sink. In addition, the different sojourn times dictated by the different schemes result in different trade offs between energy consumption and latency (see Fig. 3 and Table III). In the STATIC setting the sink is statically placed in the center of the deployment area, where the average length of the routing paths is minimized. This explains the lower average packet latency. Given its stochastic nature, the RM heuristic positions the sink mostly at the center of the deployment area. This results on one side in a lower balancing of energy consumption, but also in lower average data latency. The PM heuristic forces the sink to move on the perimeter of the deployment area. This results in a high average data latency. GMRE latency is comparable to that of PM. In the considered scenario GMRE makes the sink sojourn mostly at the corners and borders of the deployment area. However, the different distribution of the sojourn times makes GMRE more performing in terms of network lifetime with respect to PM. For what concerns latency and network lifetime, OPT performs better than PM and GMRE. Its average data packet latency is comparable to that of RM.

We notice that despite there are noticeable differences among data latency in the four proposed sink mobility schemes, these latencies are *all* extremely low with respect to the latencies observed in the data MULEs scenarios, which are four to five orders or magnitude higher.

Another message that results clear from our detailed investigation is that the best sink movement strategy depends on the considered scenario. When performing experiments in different situations (e.g., different traffic pattern, nodal deployment, data routing, and node transmission range) we have observed that staying at the corners and on the perimeter does not always translate into higher network lifetime, and, in general, in better performance. Approaches such as RM (PM) fail to capture those dynamics of the whole sensor networks that pertain to crucial aspects (data routing costs, residual energy, etc.) making the sink sojourn in the central area (on the perimeter) independently of the specific network condition and scenario. On the other hand, OPT and GMRE drive the sink to those areas (included the central sites, when convenient) that are best from a residual energy point of view.

Overall, GMRE shows a good tradeoff between network lifetime gains and latency increases, promising to be a suitable solution for those application scenarios that cannot be covered by the data MULEs mechanism.

### *C. Data MULEs vs. GMRE*

The following Table IV gives a bird-eye view of the results of the comparison between GMRE and the data MULE solutions.

TABLE IV

GMRE AND DATA MULES, GENERAL COMPARISON

| | GMRE | Data MULEs |
|---|---|---|
| Data latency | Low | High |
| Energy consumption | Medium | Low |
| Packet delivery ratio | High | Medium |
| Computational needs | Medium | Low |

In case of low network traffic and of limited deployment area size (such as those we considered for comparing the performance of the two different paradigms) we observed that both GMRE and the data MULEs deliver all generated packets to the sink. In general, successful packet delivery for low/medium network traffic is more problematic for the data MULEs solution than for the multi-hop approach (GMRE). As the deployment area grows in size, the inter-arrival time of a MULE at the same cell also grows so that overflow of sensor nodes queue may occur, resulting in a degradation of the packet delivery ratio. GMRE is instead always able to deliver packets to the sink until network congestion occurs.

The advantage of using an approach to mobility à la data MULEs becomes evident when energy consumption and sensor node computational needs are concerned. Given the single-hop nature of data exchange between the sensors and the MULEs, nodes are not required to implement a full protocol stack. Basic physical and MAC layer functions are sufficient for all data communication needs. This translates into lower energy consumption at the sensor nodes as well as lower sensor node complexity (and therefore nodal cost). In the basic scenario described above (361 nodes) sensor nodes in a data MULEs setting consume one order of magnitude less energy than in the multi-hop scenario. As expected, this gain in energy conservation is heavily paid in terms of end-to-end data latency. As mentioned, a difference as high as four orders of magnitude is observed between the two considered mobility solutions! This is caused by the extremely long time it takes to a MULE for visiting the same cell twice as well as the time needed to go back to the sink once a cell is visited. These two factors are of course dependent on the number of MULEs, their speed as well as on the size of the deployment area. Even varying MULEs speed from pedestrian (as shown here) to slow vehicular we would not be able to observe improvements of more than one order of magnitude. Only incurring the cost of introducing a high number of data MULEs would satisfy the latency requirements of many WSN applications.

## V. CONCLUSIONS

The paper explored ways of using the mobility of resource-rich network component to improve the performance of WSNs. In particular, we have investigated the performance of two WSN mobility paradigms, namely, the data MULEs solution where data routing is single-hop, and the solutions with multi-hop routing to a mobile sink. Our comparative performance evaluation highlights pros and cons of the two approaches and shows how controlled mobility is effective in prolonging network lifetime while containing the packet end-to-end delay, which is usually very high in situation when the mobility is uncontrolled.

## REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sanakarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, March 2002.

[2] F. Zhao and L. Guibas, *Wireless Sensor Networks, An Information Processing Approach*, ser. Networking. San Francisco, CA: Morgan Kaufmann, January 2004.

[3] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: Modeling a three-tier architecture for sparse sensor networks," in *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, SNPA 2003*, Anchorage, AK, May 11 2003, pp. 30–41.

[4] S. Basagni, A. Carosi, E. Melachrinoudis, C. Petrioli, and M. Z. Wang, "Protocols and model for sink mobility in wireless sensor networks," *ACM Mobile Computing and Communication Review, $MC^2R$*, vol. 10, no. 4, pp. 28–30, October 2006.

[5] ——, "Controlled sink mobility for prolonging wireless sensor networks lifetime," *ACM/Elsevier Wireless Networks*, 2007, to appear.

[6] S. Jain, R. C. Shah, W. Brunette, G. Borriello, and S. Roy, "Exploiting mobility for energy-efficient data collection in sensor networks," in *Proceedings of the IEEE Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, WiOpt 2004*, Cambridge, UK, March 24–26 2004.

[7] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke University, Durham, NC, Tech. Rep. CS-200006, April 2000.

[8] J. Luo and J.-P. Hubaux, "Joint mobility and routing for lifetime elongation in wireless sensor networks," in *Proceedings of IEEE Infocom 2005*, vol. 3, Miami, FL, March 13–17 2005, pp. 1735–1746.

[9] "ASH transceiver designer's guide," www.rfm.com, May 19 2004.