Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks

Rahul C. Shah
Intel Research
Seattle, WA 98105

Sumit Roy
Intel Corp.
Hillsboro, OR 97124

Sushant Jain, Waylon Brunette
University of Washington
Seattle, WA 98105

***Abstract*— This paper presents and analyzes an architecture that exploits the serendipitous movement of mobile agents in an environment to collect sensor data in sparse sensor networks. The mobile entities, called MULEs, pick up data from sensors when in close range, buffer it, and drop off the data to wired access points when in proximity. This leads to substantial power savings at the sensors as they only have to transmit over a short range. Detailed performance analysis is presented based on a simple model of the system incorporating key system variables such as number of MULEs, sensors and access points. The performance metrics observed are the data success rate (the fraction of generated data that reaches the access points) and the required buffer capacities on the sensors and the MULEs. The modeling along with simulation results can be used for further analysis and provide certain guidelines for deployment of such systems.**

I. Introduction

Advances in device technology, radio transceiver designs and integrated circuits along with evolution of simplified, power efficient network stacks have enabled the production of small and inexpensive wireless sensor devices [1], [2], [3], [4]. These small and inexpensive devices can be networked together to enable a variety of new applications that include environmental monitoring, seismic structural analysis, data collection in warehouses, traffic analysis etc. Such networks should collect data (typically infrequently) from the sensors for long periods of time without requiring human intervention. The sensors must be low in cost and work within a limited energy budget. Therefore, in order to achieve network longevity, a primary concern in such networks is power management.

Depending upon the application, sensors may need to be spread over a large geographical area resulting in a *sparse* network. The sensor distribution can be homogeneous (uniform spread of sensors) or heterogeneous (islands of sensors separated by large distances). Sensors at each city intersection are an example of a homogeneous distribution while sensors for habitat monitoring [5] are distributed heterogeneously. Possible approaches to ensure connectivity in such sparse networks include:

- Installation of multiple base stations to relay the data from sensor nodes in their coverage area.
- Deploying enough low-power relay nodes to effectively form a *dense* connected network [6].

The base station approach trades off high communication power needed by the sensors (for fewer BSs) with the cost of installing additional stations. On the other hand, deploying nodes to form a dense, fully-connected ad-hoc network may not be cost-effective either. The proposed architecture in this paper seeks to retain the advantages of both approaches - i.e. achieve cost-effective connectivity in sparse sensor networks while reducing the power requirements at sensors.

The key to making this feasible is the ubiquitous existence of mobile agents [7] in many of our target scenarios that we term MULEs (Mobile Ubiquitous LAN Extensions) [8]. In the case of traffic monitoring, this role is served by vehicles (cars, buses) outfitted with transceivers; in a habitat monitoring scenario, animals can perform this role. MULEs are assumed to be capable of *short-range* wireless communication and can exchange data from any nearby sensor or access point they encounter as a result of their motion. Thus MULEs can pick up and buffer data from the sensors when they are close, transferring the data to an access point when it comes within range.

The primary advantage of our approach is the large power savings that occur at the sensors because communication now takes place over a short-range. Promising new radio technologies like Ultra-Wideband (UWB) [9] which operate at extremely low-power with large burst data capacity are potentially suited for sensor to MULE communication. The primary disadvantage of this approach, however, is increased latency because sensors have to wait for a MULE to approach before the transfer can occur. Nevertheless for many data collection applications (that require data for analysis purposes only on the order of hours or even a day) such increased latency is acceptable. The proposed three-tier MULE architecture is thus suitable for such delay-tolerant scenarios where power budgets at the sensor are the over-riding constraint.

The relative strengths and weaknesses of various approaches for data collection in sparse sensor networks are qualitatively summarized in Table I. In the base station approach there are a few base stations (same as access points) that cover the entire geographical area and each sensor communicates directly with the nearest base station. In the ad hoc network approach, enough relay nodes are present so as to form an ad hoc network among sensors and relay nodes. The sensors then send their data to the wired access points by multi-hop routing over this ad hoc network. Note that while the MULE approach suffers from higher latency, it has both low sensor power consumption and low infrastructure cost; characteristics that may be important for many applications.

	Performance Metrics			
Approaches	Latency	Sensor Power	Data Success Rate	Infrastructure Cost
Base Stations	Low	High	High	High
Ad-hoc network	Medium	Medium-low	Medium	Medium-high
MULE	High	Low	Medium	Low

TABLE I

PERFORMANCE OF DIFFERENT APPROACHES FOR DATA COLLECTION IN SPARSE WIRELESS SENSOR NETWORKS.

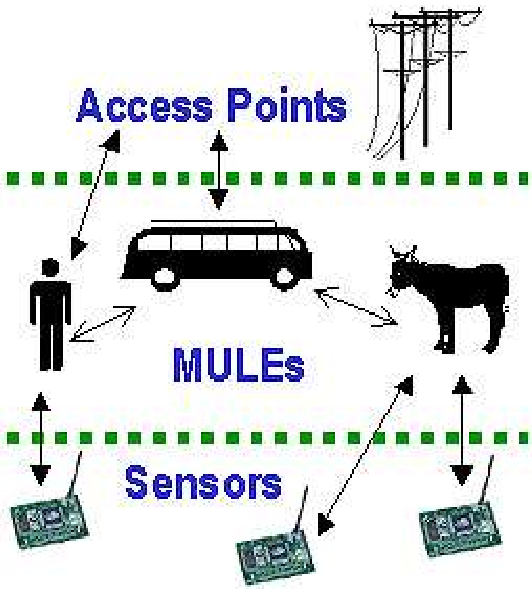

Fig. 1. The MULEs three-tier architecture

The next section presents the MULEs architecture in more detail. After that, the rest of the paper focuses on modeling the system to obtain initial insights into the performance of such an architecture. The goal of modeling was to understand the scaling of the system characteristics as the parameters - number of sensors, MULEs etc. change. The model chosen was very simple, which enabled us to obtain closed form analytical results for many quantities of interest, including data success rate (the fraction of generated data that reaches access points) and buffer occupancies at MULEs and sensors. Although latency is an important performance metric, it is not analyzed in this paper due to lack of space. In addition to detailing the analysis, system simulation results are also presented. These verify the analysis while providing some more insight into system performance. The paper finally concludes with the insights gained from the modeling analysis and simulation results and outlines future research directions based on this initial work.

II. THE MULE THREE-TIER ARCHITECTURE

The MULE architecture provides wide-area connectivity for a sparse sensor network by exploiting mobile agents such as people, animals, or vehicles moving in the environment. The system architecture comprises of a three-tier layered abstraction (Fig. 1) that can be adjusted to different types of situations and distribution needs:

- A top tier of WAN connected devices,
- A middle tier of mobile transport agents and
- A bottom tier made of fixed wireless sensor nodes.

The top tier is composed of access points/central repositories, which can be set up at convenient locations where network connectivity and power are present. These devices communicate with a central data warehouse that enables them to synchronize the data that they collect, detect duplicates, as well as return acknowledgments to the MULEs (acks may be necessary to ensure reliability of data for certain applications).

The intermediate layer of mobile MULE nodes provides the system with scalability and flexibility for a relatively low cost. The key traits of a MULE are large storage capacities (relative to sensors), renewable power, and the ability to communicate with the sensors and networked access points. MULEs are assumed to be serendipitous agents whose movements cannot be predicted in advance. However as a result of their motion, they collect and store data from the sensors, as well as deliver acks back to the sensor nodes. In addition, MULEs can communicate with each other to improve system performance. For example a multi-hop MULE network can be formed to reduce the latency between MULE and access point.

The bottom tier of the network consists of randomly distributed wireless sensors. Work performed by these sensor nodes should be minimized as they have the most constrained resources of any of the tiers.

Depending on the application and situation, a number of tiers in our three-tier abstraction could be collapsed onto one device. For example, to reduce latencies in the traffic monitoring application, the MULEs can be equipped with an always-on connection (such as a cellular or satellite phone) which would allow it to act as the top and the middle tier.

Another key advantage of the MULE architecture is its scalability and robustness as compared to centralized solutions. No sensor depends on any single MULE, and hence failure of any particular MULE does not disconnect the sensor from the sparse network. It only degrades the performance. Also the MULE architecture is easily scalable as deployment of new sensors or MULEs requires no network configuration and (most importantly) obviates the need for algorithmic scalability for key functions such as routing of packets. The three-tier architecture supports increased reliability as the redundant access points and multiple MULEs create a fault-tolerant design wherein failures merely lead to graceful degradations (reduced data success rate and increased latency).

Acknowledgments are an option in the MULE architecture; as mentioned earlier, they may be essential to ensure reliability for certain applications. We chose to use an end-to-end acknowledgment system to ensure robustness to MULE failures. Another possibility was to have tier-to-tier reliability where the MULEs ack the sensor and the access points ack the MULEs in turn. This was avoided, however, as it is possible that the MULEs may fail at any time without delivering the data to the access points and also because the MULEs may not be trusted agents (data sent by the sensors may also be encrypted for this reason). Furthermore, we chose to use a cumulative acking scheme to keep the number of transmissions between MULEs and sensors smaller as well as to reduce the overall storage requirements on the MULEs.

In summary, the benefits of our system include:

- Far less infrastructure than a fixed base-station approach. For applications with few sensors spread over a large area, the cost savings could be orders of magnitude.
- There is no overhead associated with routing packets from other sensors as compared to an ad hoc network approach. For large ad hoc networks, this overhead can lead to a substantial increase in energy consumption at a node.
- Given a sufficient density of MULEs, the system is more robust than a traditional fixed network. Since sensors only rely on MULEs, and MULEs are interchangeable, the failure of any number of MULEs does not mean connectivity failure; it merely increases the latency and decreases the data success rate of the network.
- The endpoint applications in this transaction need not be aware of the mobile portions of the network. Our network API offers a packet-level reliable transport protocol. To both the sender and receiver, it appears as if they are talking over a reliable network with large and highly variable latencies.
- System flexibility allows the same transport medium to be used simultaneously by different applications. The MULE system can be viewed as a mobile transport mechanism for connecting heterogeneous nodes.

The drawbacks of our system are:

- Latency for this type of network is high and limits the types of applications this solution would be applicable for. Deterministic delay bound guarantees seem feasible only if MULEs traverse fixed routes.
- The system presupposes a sufficient amount of physical movement in the environment, which is a property of many sensor systems.
- While no network is guaranteed to successfully deliver data all the time, our serendipitous network can encounter unexpected failures such as loss of a MULE or inability to reach sensors because of change in terrain causing limitations in mobility.

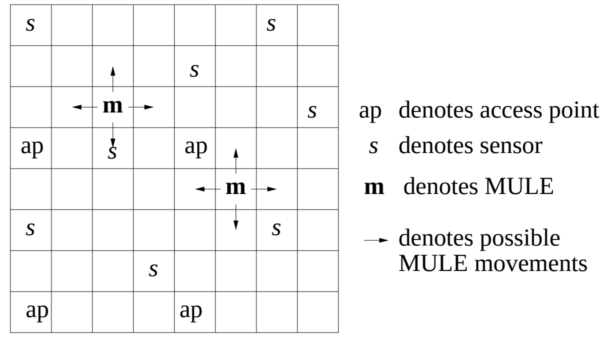

Fig. 2. A two dimensional grid with the different system components

III. System Modeling

We now focus on a simple and fully discrete (in time and space) model of the network that nevertheless allows us to investigate system performance as the parameters are scaled. Figure 2 shows a pictorial representation of different system components. We make the following assumptions in our modeling:

- The underlying topology on which sensors, MULEs and access points are placed is assumed to be a discrete and finite two-dimensional grid. Further, for analytical simplicity the planar topology is assumed to be the surface of a torus (i.e the grid is wrapped in both the north-south and the east-west direction).
- Only a fraction of the grid points are occupied by sensors and access points. The access points are modeled to be uniformly spaced on the grid while the sensors are randomly distributed.
- The network evolves synchronously with a global clock. At every clock tick the following events take place:
  - Sensors generate one unit of data
  - Every MULE moves on the grid
- The MULE motion is modeled as a simple symmetric random walk on the grid. At every clock tick, a MULE moves with equal probability to any of the four neighbors of its current grid position.
- The MULEs communicate with the sensors or access-points only when they are co-located at the grid points; the communication is error-free and all the data is assumed to be transferred during the contact period. Our model therefore assumes sufficiently large link capacities.
- MULEs move independent of each other and do not exchange any data among themselves when they intersect (occupy the same grid point)
- Both sensors and MULEs have buffers to store data. For the sensors, generated data is placed in its buffer if it has space otherwise the new data is dropped. Similarly any data transferred from sensors to MULEs is placed in the MULE buffer only if space is available, else it is dropped. Initially all buffers are empty.
- There are no end-to-end acknowledgments to ensure the reliable delivery of data.

Based on this model, we analyze the performance of the system as the grid size, number of access points and the number of MULEs are changed. The performance measures that we focus on are:

- Data Success Rate: It measures the fraction of generated data at the sensors that the system is able to transfer to the access-points. In an ideal system all the data generated by the sensors would be transferred to the access-points. This would yield a data success rate of one.
- Buffer Sizing: As mentioned earlier both sensors and MULEs have buffers. While small buffers could lead to high packet drop rates, reducing the data success rate, large buffers have an associated penalty in terms of energy consumption, physical size and manufacturing costs. Thus we would like to determine mimimum buffer sizes that would ensure high data success rate while being cost-effective.

The model presented above is very simple and excludes many real-world aspects such as radio propagation, link failure and bandwidth constraints. Another major concern is the choice of mobility model for analysis. We realize that a discrete random walk is not an accurate representation of the motion of vehicles, people etc. However, the simplicity of this model enables us to obtain closed-form results for the quantities of interest, giving us insight into system scalability. Also as mentioned in a recent survey [10], random walk is a widely used mobility model which is useful in modeling the unpredictable motion of entities. We hope to incorporate more advanced and realistic models such as Smooth Random Mobility Model [11] or Brownian motion with drift [12], [13] in subsequent analysis. However note that with the increasing complexity of mobility models the hope of closed form analysis diminishes and one has to rely primarily on simulation. Thus we believe that a first order analysis with our simple model provides us with a useful base.

IV. Glossary of Notation and Symbols

This section lists all the commonly used symbols and notation in this paper:

$(X_n)_{n\geq 0}$	A discrete-time Markov chain
$\mathcal{S}$	State space of the Markov chain
p_{ij}	The transition probability $P\{X_{n+1} = j \vert X_n = i\} \forall i,j \in \mathcal{S}$
π=$(\pi_i : i \in \mathcal{S})$	Stationary distribution for the Markov chain
$\vert A \vert$	The cardinality of a set A
N	The number of points on the grid, i.e. the grid is $\sqrt{N}$ on a side
N_{mules}	The number of MULEs in the system
N_{AP}	The number of access points (AP) in the system
$N_{sensors}$	The number of sensors in the system
ρ_{mules}	The ratio of the number of MULEs to the grid size (N_{mules}/N); $(0 \leq \rho_{mules} \leq 1)$
ρ_{AP}	The ratio of the number of access points to the grid size (N_{AP}/N); $(0 \leq \rho_{AP} \leq 1)$
$\rho_{sensors}$	The ratio of the number of sensors to the grid size $(N_{sensors}/N)$; $(0 \leq \rho_{sensors} \leq 1)$
MB	The total buffer capacity on a MULE (in number of packets)
SB	The total buffer capacity on a sensor (in number of packets)
AP	Access point
H_i	The hitting time to a sensor i in the grid, i.e. the time taken by a MULE starting from the stationary distribution to first hit i when there is only one MULE in the system
R_i	The inter-arrival time to a sensor i in the grid, i.e. the time between consecutive MULE arrivals to i when there is only one MULE in the system
$H_i^{N_{mules}}$	The hitting time to a sensor i in the grid by any mule when there are N_{mules} in the system
$R_i^{N_{mules}}$	The inter-arrival time at a sensor i in the grid by any mule when there are N_{mules} in the system
R_{AP}	The time taken by a particular MULE to start from the set of access points and return back to it
Z_i	The buffer occupancy for a sensor i with $SB = \infty$ when a MULE visits it
$M^{(k)}$	The buffer occupancy for MULE k on one excursion from the set of access points back to the set. If there is only one MULE, then we'll drop the superscript for convenience
S	The data success rate of the system, which is the fraction of generated data that reaches the access points

V. Basic Results

The most elementary reference scenario consists of one access point $(N_{AP} = 1)$ and one MULE $(N_{mules} = 1)$ in the system. We assume that the MULE and the sensors have infinite buffer capacity. The AP is at some position (the exact position is not critical) in the grid of size $\sqrt{N}$ on a side. The MULE is assumed to perform a simple symmetric random walk on the grid. The state space $\mathcal{S}$ consists of the points on the grid scanned in any order to form a vector of length N (i.e., $|\mathcal{S}| = N$). This simple model allows us to apply the large body of relevant results from discrete-time, finite state Markov chains. We rely on the stationary distribution $\pi = (\pi_i : i \in \mathcal{S})$ to estimate average values of the quantities of interest.

The transition probabilities for the Markov chain with state space $\mathcal{S}$ are:

$$p_{ij} = \begin{cases} 1/4 & \text{if (i,j) has an edge} \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

Since $\sum_{i \in \mathcal{S}} \pi_i = 1$ and all states are equiprobable (i.e $\pi_i = \pi_j \ \forall i,j \in \mathcal{S}$), we get,

$$\pi_i = \frac{1}{N} \qquad (2)$$

We next compute the following:

- Average inter-arrival time at a sensor node i, $E[R_i]$
- Average length that the MULE traverses before it returns to the AP, $E[R_{AP}]$
- Average number of data samples the MULE picks up during one traversal, $E[M]$

The average time it takes for the MULE to return to the same sensor node i is the inverse of the stationary probability by Markov chain theory. Therefore,

$$E[R_i] = \frac{1}{\pi_i} = N \tag{3}$$

Since a unit data is generated every clock tick, this is also the average value of the buffer occupancy at the sensor $E[Z_i]$ *when* the MULE visits it (because $SB = \infty$, so the buffer occupancy is the same as the amount data generated). Note that this is the average value of the sensor buffer occupancy observed only when the MULE visits the sensor, not over all instants of time (the second quantity is not of much use in analyzing the system and is also harder to characterize).

Similarly, the average number of steps the MULE takes before returning to the access point is:

$$E[R_{AP}] = \frac{1}{\pi_{AP}} = N \tag{4}$$

The number of data samples the MULE picks up during one traversal depends on three things - the length of the traversal R_{AP}, number of sensors encountered which depends on $\rho_{sensors}$ and the buffer occupancy at the sensors Z_i. Since the three quantities are independent, the average is simply given by (since $MB = \infty$),

$$\begin{aligned} E[M] &= E[R_{AP}] \cdot \rho_{sensors} \cdot E[Z_i] \\ &= E[R_{AP}] \cdot \rho_{sensors} \cdot E[R_i] \\ &= \rho_{sensors} N^2 \end{aligned} \tag{5}$$

The above results provide useful preliminary insights into the performance of the system as the grid is scaled. Clearly, the time between MULE visits to a sensor grows linearly with the grid size as shown in (3). This has two implications. Firstly, the required buffer at the sensor needs to scale with the grid size to prevent loss of data[1]. Secondly, the latency for data samples also increases with the grid size. Both these problems can be mitigated by having multiple MULEs in the system, a case considered in section VII.

The second insight is that with only one access point in the system, the length of MULE excursions from the AP to the AP grows linearly as shown in (4). Similar to the case above, there are two implications. The first is that the required MULE buffer needs to be large to prevent loss of data. In fact, the required buffer size grows as the square of the grid size as shown by (5) above (Again we use $E[M]$ to get an idea of the buffer sizes needed to avoid packet drops). The second implication is that the latency for the data when traveling from the sensor to the access points grows linearly. This means that the number of access points in the system needs to scale with the grid size, a case considered in section VI.

[1]Notice that while we assume $SB = \infty$, in reality the buffer capacity has to be finite but sufficiently large to avoid packet drops. Thus we use $E[R_i]$ to provide an indication of *sufficiently large*.

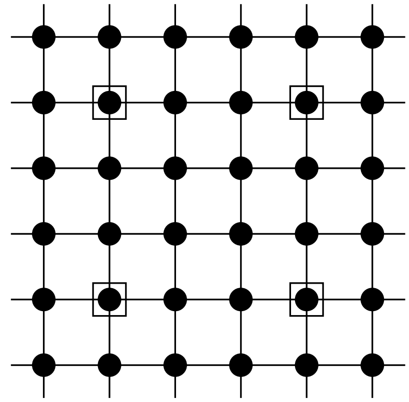

Fig. 3. A two dimensional grid with the squares representing the positions of the access points

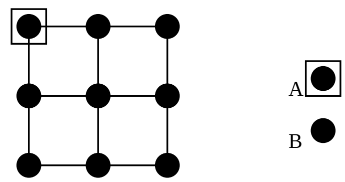

Fig. 4. Folded version of the two dimensional grid to form a smaller grid (The types of nodes and their transition probabilities are also shown)

VI. Scaling with Number of Access Points

In this section, we analyze the effect of multiple access points in the system. We assume that the access points are spaced at a distance of $\sqrt{K}$ points on the grid in both the x and y directions. Therefore, $K = N/N_{AP} = 1/\rho_{AP}$. We still assume that only one MULE is present in the system.

Result 1: If the access points are regularly spaced at a distance of $\sqrt{K}$ points on the grid in both the x and the y directions, then the expected length of excursion for the MULE starting from the set of access points till it reaches the set again (could be the same AP or another one),

$$\begin{aligned} E[R_{AP}] &= K \\ &= \frac{1}{\rho_{AP}} \end{aligned} \tag{6}$$

Proof: Looking at the symmetry of the grid in Fig. 3, we can reduce the state space to a smaller grid of size $\sqrt{K} \times \sqrt{K}$ as shown in Fig. 4. This can be seen to be the result of folding the entire grid onto the smaller box containing only one access point A (which represents all the access points). This is possible because from the perspective of a MULE, all access points are equivalent. The resultant grid also remains a torus (wraps around in the north-south and east-west directions).

As in section V the stationary distribution for a node i in this reduced grid (size $\sqrt{K} \times \sqrt{K}$) can be shown to be:

$$\pi_i = \frac{1}{K} \tag{7}$$

Using this stationary distribution, the return time to the point "A" can be calculated. This is also the required excursion time of the MULE from the AP set to the AP set since the point "A" represents all the access points of the original grid.

$$\begin{aligned} E[R_{AP}] &= \frac{1}{\pi_A} \\ &= K \\ &= \frac{1}{\rho_{AP}} \end{aligned}$$

■

Thus we see that the MULE excursion length between the access point set is independent of the grid size as long as the number of access points scale as a fraction of the grid size.

VII. Scaling with number of MULEs

In this section, we analyze the case when there are multiple MULEs in the system. The fraction of MULEs in the system is kept constant as the size of the grid is increased, i.e., $N_{mules}/N = \rho_{mules}$. We first calculate the average number of visits observed at a sensor per unit time. We then calculate the expected inter-arrival times for MULEs to a sensor. That will extend the result (3) obtained in section V. As mentioned before, we assume that all the MULEs are performing independent random walks, with no communication among each other. Also, note that every MULE starts in the stationary distribution, and subsequently performs a random walk, thus remaining in the stationary distribution.

Now consider a sensor and a particular MULE M_0. Then the probability that M_0 intersects the sensor is given by,

$$P\{M_0 \text{ intersects sensor}\} = \frac{1}{N} \tag{8}$$

Define:

$$Y_k = \begin{cases} 1 & \text{if one or more MULEs intersects the sensor at time k} \\ 0 & \text{if no MULE intersects the sensor at time k} \end{cases} \tag{9}$$

Hence the probability that no MULE intersects with the sensor is given by,

$$\begin{aligned} P\{Y_k = 0\} &= \left(1 - \frac{1}{N}\right)^{N_{mules}} \\ \Rightarrow P\{Y_k = 1\} &= 1 - \left(1 - \frac{1}{N}\right)^{N_{mules}} \end{aligned} \tag{10}$$

Therefore the expected number of MULE visits to a sensor per unit time[2] is,

$$\begin{aligned} \lim_{n\to\infty} \frac{1}{n} E\left[\sum_{k=0}^{n-1} 1_{\{Y_k=1\}}\right] &= \lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} P\{Y_k = 1\} \\ &= 1 - \left(1 - \frac{1}{N}\right)^{N_{mules}} \\ &\approx 1 - e^{-\rho_{mules}} \quad (\text{large } N) \quad (11) \\ &\approx \rho_{mules} \quad (\text{small } \rho_{mules}) \quad (12) \end{aligned}$$

Result 2: The average inter-arrival time between MULE visits to a sensor i when there are N_{mules} in the system is given by,

$$\begin{aligned} E[R_i^{N_{mules}}] &= \frac{1}{1 - (1 - \frac{1}{N})^{N_{mules}}} && (13) \\ &\approx \frac{1}{1 - e^{-\rho_{mules}}} \quad (\text{large } N) && (14) \\ &\approx \frac{1}{\rho_{mules}} \quad (\text{small } \rho_{mules}) && (15) \end{aligned}$$

Proof: To find the average inter-arrival time at a sensor i, we consider the Markov chain composed of the product of the Markov chains of each of the MULEs. Thus the new state space is given by,

$$S' = \underbrace{S \times S \times \ldots \times S}_{N_{mules} \text{times}}$$

In the modified state space S', we are interested in the set of states A which represent one or more MULEs intersecting i. Since all the states are equally likely, the stationary distribution for the set A can be calculated as,

$$\begin{aligned} \pi(A) &= \frac{|A|}{|S'|} \\ &= \frac{|S'| - |S' - A|}{|S'|} \\ &= \frac{N^{N_{mules}} - (N-1)^{N_{mules}}}{N^{N_{mules}}} \\ &= 1 - (1 - \frac{1}{N})^{N_{mules}} \end{aligned} \tag{16}$$

Thus, using Kac's formula [14], the average inter-arrival time between MULE visits to a sensor i is,

$$\begin{aligned} E[R_i^{N_{mules}}] &= \frac{1}{\pi(A)} \\ &= \frac{1}{1 - (1 - \frac{1}{N})^{N_{mules}}} \end{aligned}$$

■

Corollary 2.1: Average buffer occupancy on a sensor[3] (with sufficiently large buffer capacity) can now be calculated as:

$$\begin{aligned} E[\text{Sensor Buffer}] &= E[R_i^{N_{mules}}] \\ &\approx \frac{1}{\rho_{mules}} \end{aligned} \tag{17}$$

[2]Multiple MULEs intersecting the sensor at the same time is considered to be just one intersection.

[3]As mentioned earlier, this is just the average buffer occupancy seen at the times of MULE arrivals at the sensor; not at all times. In other words, this is the average amount of data that needs to be picked up by a MULE at the sensor.

Here we have used the observation that the sensor buffer occupancy at the times of MULE visits is exactly the same as the inter-arrival times between MULEs. Hence the average values are also the same.

Corollary 2.2: Average buffer occupancy on a MULE[4] (with sufficiently large buffer capacity) can also be calculated as:

$$\begin{aligned} E[\text{Mule Buffer}] &= \rho_{sensors} E[R_{AP}] E[R_i^{N_{mules}}] \\ &\approx \frac{\rho_{sensors}}{\rho_{AP}\rho_{mules}} \end{aligned} \tag{18}$$

Similar to the previous corollary, we use the expected value of the inter-arrival times at a sensor as the expected value of the sensor buffer occupancy when a MULE visits it.

It is interesting to note that the problem of increasing buffer requirements at the sensor as the grid increases which we encountered in section V is eliminated. As long as ρ_{mules} remains constant, the buffer requirements remain the same. So far we have just found the average value of the inter-arrival times for MULEs to a sensor. We next need to obtain the probability distribution. However, we first find the probability distribution for the hitting time at a sensor as that is needed for the result on the inter-arrival times.

A. Hitting time distribution at a sensor

For our purposes, the hitting time for a sensor i is defined as the first time a MULE hits i when all the MULEs start from the stationary distribution. We first find the probability distribution of the hitting time for a system with a single MULE before evaluating the general case of multiple MULEs. [14] shows that the mean of the hitting time for a single MULE is $\Theta(N \log N)$ for simple symmetric random walk on the surface of a torus. Furthermore, the distribution of hitting times for an ergodic Markov chain can be approximated by an exponential distribution of the same mean [14]. Therefore,

$$P\{H_i > t\} \approx \exp\left(\frac{-t}{cN \log N}\right) \tag{19}$$

where the constant $c \approx 0.34$ as $N \to \infty$ (valid for $N \geq 25$) [15]. Note that this result uses the continuous time version of the discrete time Markov chain, but the result is still correct for the discrete time case [14]. However, writing in continuous time simplifies the analysis considerably, thus all the hitting and return time probability distribution results will be for the continuous time chain. Using this we can now extend the result for the case when there are $N_{mules}(> 1)$ in the system.

Result 3: The hitting time for a sensor i when there are N_{mules} in the system, all of which start in the stationary distribution is given by:

$$P\{H_i^{N_{mules}} > t\} \approx \exp\left(\frac{-t}{0.34\frac{N}{N_{mules}}\log(N)}\right) \tag{20}$$

[4]Similar to the sensor buffer occupancy, this is the average buffer occupancy on the MULE as seen at the times of MULE intersections with an AP; not at all times. Thus this is the average amount of data that is picked up by the MULE during one excursion between the AP set.

Proof: Let $H_i^{(k)}$ denote the hitting time to sensor i for a single MULE k. Then,

$$H_i^{N_{mules}} = \min_{k \in MULEs} H_i^{(k)} \tag{21}$$

Thus, we obtain,

$$\begin{aligned} P\{H_i^{N_{mules}} > t\} &= [P\{H_i > t\}]^{N_{mules}} \\ &\approx \left[\exp\left(\frac{-t}{0.34N\log(N)}\right)\right]^{N_{mules}} \\ &= \exp\left(\frac{-t}{0.34\frac{N}{N_{mules}}\log(N)}\right) \end{aligned}$$

■

B. Inter-arrival time distribution at a sensor

To find the inter-arrival time distribution at a sensor i, we first consider the case when there is only one MULE in the system. In that case, the inter-arrival time at i is the same as the return time R_i for the MULE. Unfortunately, there is no closed form result for the distribution, but can only be approximated as $\pi/\log t$ for $t \to \infty$ for an infinite grid [16]. For smaller times and for finite grid sizes, this only provides a very loose upper bound on the tail probability.

To obtain a better characterization we derive a recursive equation to compute $P\{R_i = t\}$ (inter-arrival time distribution for a single MULE). Let the initial position of the MULE be at the grid position 0. Define $L_{i,j}(t)$ to be the number of paths starting from i and ending at j of length t, avoiding the point 0 at all the intermediate steps. Also, let the neighbors of a node k in the torus be denoted by the set $\mathcal{N}(k)$. Then, without loss of generality, for any sensor node i,

$$P\{R_i = t\} = L_{0,0}(t)/4^t \tag{22}$$

In the above equation, $L_{0,0}(t)$ denotes the total number of valid paths that return to 0 in t steps and 4^t denotes the total number of possible paths of t steps. The following recursive equation can now be used to compute $L_{0,0}(t)$:

$$\begin{aligned} L_{i,j}(t) &= \textstyle\sum_{k \in \mathcal{N}(i) \wedge k \neq 0} L_{k,j}(t-1), \quad t > 1 \\ L_{i,j}(1) &= \begin{cases} 1 & \text{if } j \in \mathcal{N}(i) \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

Result 4: If the number of MULEs in a system is N_{mules}, the inter-arrival time at a sensor i can be written as:

$$P\{R_i^{N_{mules}} > t\} \approx P\{H_i^{N_{mules}-1} > t\} \cdot P\{R_i > t\} \tag{23}$$

Proof: To find the inter-arrival time distribution at a sensor i, we consider the moment at which one MULE intersects the sensor[5]. At this time instant, the rest of the MULEs are in the stationary distribution. Thus,

$$R_i^{N_{mules}} = \min\left(R_i, H_i^{N_{mules}-1}\right)$$

since the MULE at the sensor has to return to the sensor, but for the $(N_{mules}-1)$ remaining MULEs, it is identical to hitting the sensors starting from stationarity. The result follows from this observation. ■

[5]Here we are ignoring multiple mules at the sensor which is a very unlikely event for low mule densities.

C. Return time distribution to the access points set

We now compute the distribution of the excursion times of a MULE between the access point set. As in section VI we consider the folded torus (Fig. 4) in which all the access points are represented as a single grid point. Since this point represents the set of all access points, we need to compute the return time distribution to this single grid point. For this we can apply (22) to the folded torus to obtain the required return time distribution. Thus,

$$P\{R_{AP} = t\} = L_{0,0}(t)/4^t \quad (24)$$

with $L_{0,0}(n)$ defined on the surface of the folded grid of Fig. 4.

VIII. Data Success Rate

We now have the pieces in place to calculate the data success rate. We define the data success rate as the ratio of the average amount of data delivered to the access points by time t to the total data generated by time t as $t \to \infty$.

Result 5: The data success rate of the system is given by,

$$S = \sum_{k \in \text{MULEs}} \frac{E\left[\min(\rho_{sensors} \sum_{i=1}^{R_{AP}} \min(R_i^{N_{mules}}, SB), MB)\right]}{E[R_{AP}]N_{sensors}} \quad (25)$$

Proof: We use renewal reward theory [17] to derive data success rate. One excursion of the MULE from the access point set back to the set is considered as a cycle. Therefore R_{AP} is the length of a cycle. Recall that the sensors generate data at the constant rate of one packet per unit time therefore the average data generated in system per unit time is $N_{sensors}$. We now get the data success rate S as,

$$S = \frac{E\left[\sum_{k \in MULEs} M^{(k)}\right]}{E[R_{AP}]N_{sensors}}$$

Here,

$$\begin{aligned} M^{(k)} &= \text{Data picked up by the MULE } k \text{ in time } R_{AP} \\ &= \min(\rho_{sensors} \sum_{i=1}^{R_{AP}} Y_i^{(k)}, MB) \end{aligned}$$

The min-function is because the buffer capacity of the MULE bounds the total amount of data a MULE can carry. Now, $Y_i^{(k)}$ is the amount of data at a sensor visited by MULE k at time i. This is given by,

$$Y_i^{(k)} = \min(Z_i, SB)$$

Similar to the previous step, the sensor buffer capacity bounds the amount of data that can be present at a sensor, hence the min-function. Also, since Z_i is the amount of data generated and not yet picked up at the sensor, it has the same distribution as the inter-arrival time at a sensor.

Hence, putting this all together,

$$S = \sum_{k \in \text{MULEs}} \frac{E\left[\min(\rho_{sensors} \sum_{i=1}^{R_{AP}} \min(R_i^{N_{mules}}, SB), MB)\right]}{E[R_{AP}]N_{sensors}}$$

■

Parameter	Description
Grid size	Number of points on the grid N
# of sensors	$= N\rho_{sensors}$
# of MULEs	$= N\rho_{MULEs}$
# of access points	$= N\rho_{AP}$
Sensor buffer size	Number of data samples each sensor can hold
MULE buffer size	Number of data samples each MULE can hold
Random seed	Used to initialize the random number generator used by the simulator

TABLE II

INPUT PARAMETERS TO THE SIMULATOR

IX. Simulation Setup

A custom event driven simulator was written to verify the preceding analysis and also explore the conditions under which it holds. In this section we present a brief description of the simulator.

The simulator is a discrete event driven simulator where time is measured in abstract units of clock-ticks. The underlying grid structure is the surface of a torus with the size N specified during initialization. Depending on the values of $\rho_{sensors}$ and ρ_{mules}, appropriate number of sensors and MULEs are placed randomly on the grid in the beginning. Buffer sizes on both the sensors and the MULEs can also be specified and are completely empty when the simulation is started. Finally, the APs can be either randomly placed on the grid or regularly spaced[6], with the number of APs depending on the value of ρ_{AP}. All the input parameters to the simulator are shown in Table II. A summary of the various events handled by the simulator is given in Table III.

The simulator also assumes a perfect radio channel, i.e., there is no loss of packets during transmission. The only way packets can be lost is if the sensor or MULE buffers overflow. However, the sensors do not maintain any state (such as acks etc.) to implement reliability. Also there is no MULE to MULE interaction, even though they may occupy the same grid point.

X. Simulation Results

In this section, simulation results are presented which verify all the major results of the analysis and also provide certain insights. While the entire parameter space cannot be explored, the simulations were performed for a range of reasonable values for the various parameters.

To verify scaling with access points, $E[R_{AP}]$ was measured for a variety of grid sizes from 25×25 to 200×200. As expected, $E[R_{AP}]$ remained constant across all grid sizes (Fig. 5) when ρ_{AP} was kept constant, verifying (6).

Fig. 6 shows the effect of scaling the number of MULEs on the average inter-arrival time to a sensor. As expected $E[R_i]$ remained constant for different grid sizes as long as the value of ρ_{mules} did not change, in accordance with (15).

Fig. 7 plots the cumulative distribution function of the hitting time $H_i^{N_{mules}}$ for $\rho_{mules} = 1\%, 10\%$ and 20% on a

[6] Interestingly, simulations showed similar results for both uniform and random placement of access points.

Event	Parameters to Event	Cause of occurrence	Action
MULE motion	MULE id	Periodic (every clock tick)	Changes the position of the MULE
Data Generation	sensor id	Periodic (every clock tick)	Generates new data at the sensor and stores it in buffer. If the sensor buffer is full, data is dropped
MULE-Sensor Interaction	MULE id, sensor id	MULE and sensor at the same grid position	Transfers all data from the sensor to the MULE. If the MULE buffer is full, all the extra data is dropped
MULE-Access Point Interaction	MULE id, access point id	MULE and access point at the same grid position	Transfers all data from the MULE to the access point

TABLE III

EVENTS DEFINED BY THE SIMULATOR

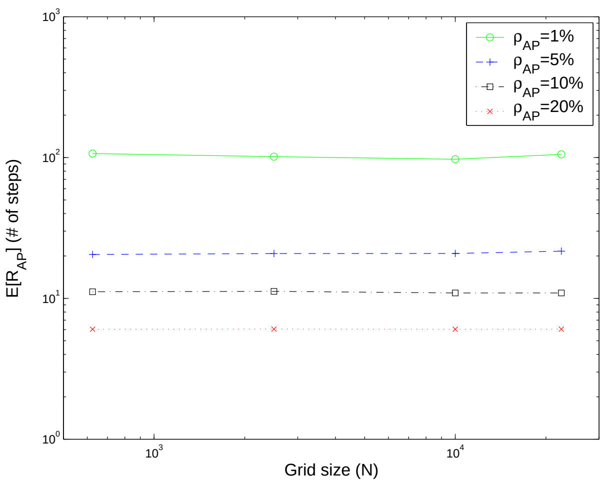

Fig. 5. $E[R_{AP}]$ while scaling the grid size with $\rho_{AP} = 1\%, 5\%, 10\%$ and 20%

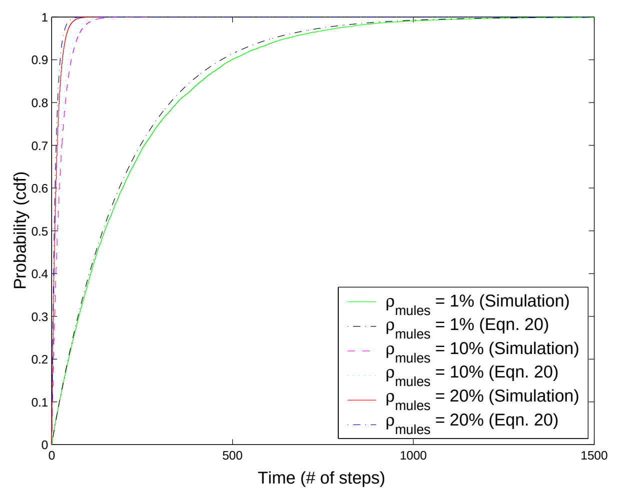

Fig. 7. Cdf of the hitting times $(H_i^{N_{mules}})$ at a sensor $(20 \times 20$ grid)

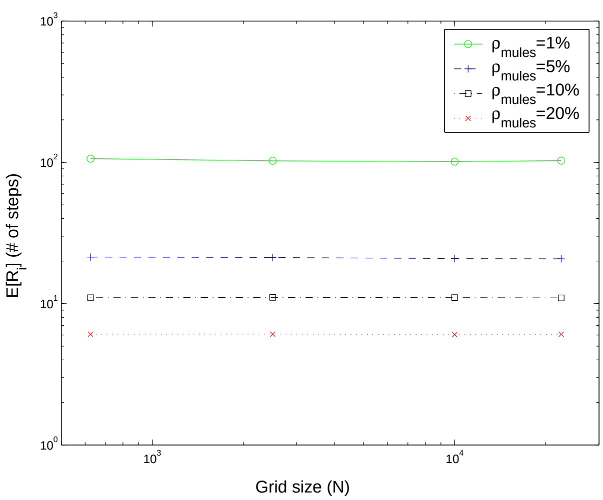

Fig. 6. $E[R_i]$ while scaling the grid size with $\rho_{mules} = 1\%, 5\%, 10\%$ and 20%

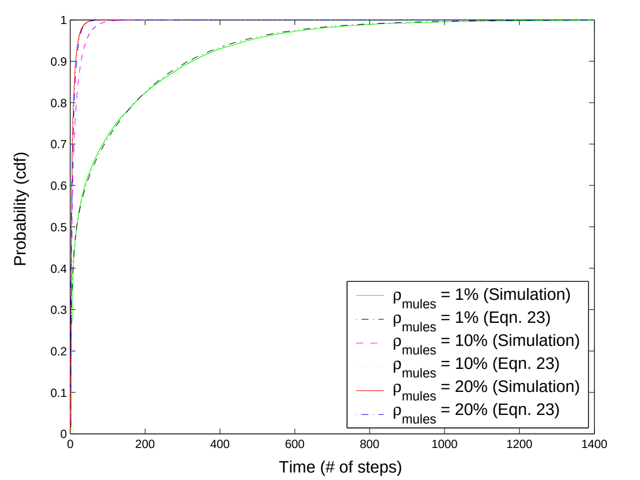

Fig. 8. Cdf of the inter-arrival times $(R_i^{N_{mules}})$ at a sensor $(20 \times 20$ grid)

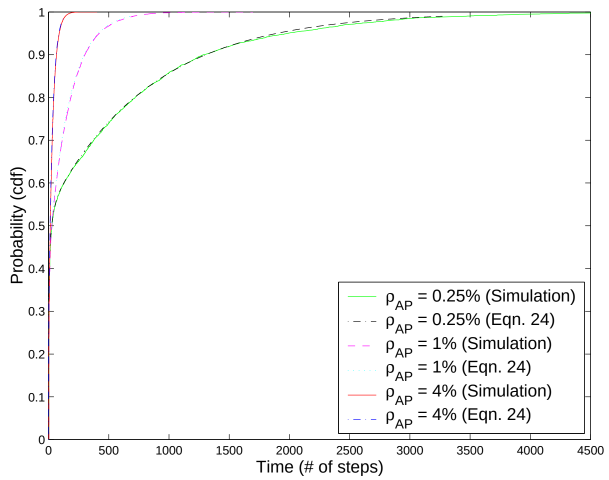

Fig. 9. Cdf of the return times (R_{AP}) for the access point set (20×20 grid)

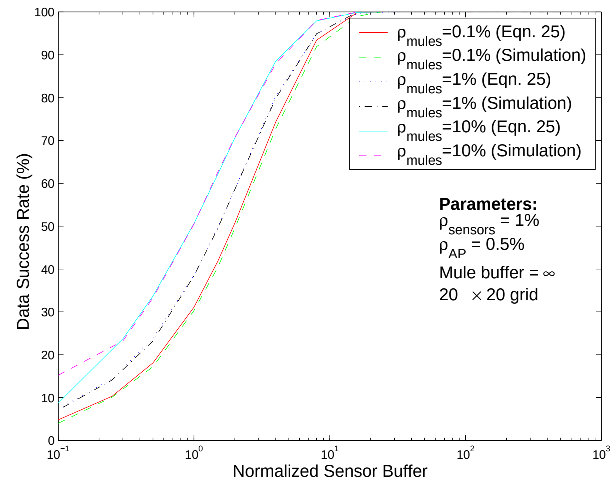

Fig. 11. Data success rate vs. normalized sensor buffer size for $\rho_{mules} = 0.1\%, 1\%$ and 10% (20×20 grid)

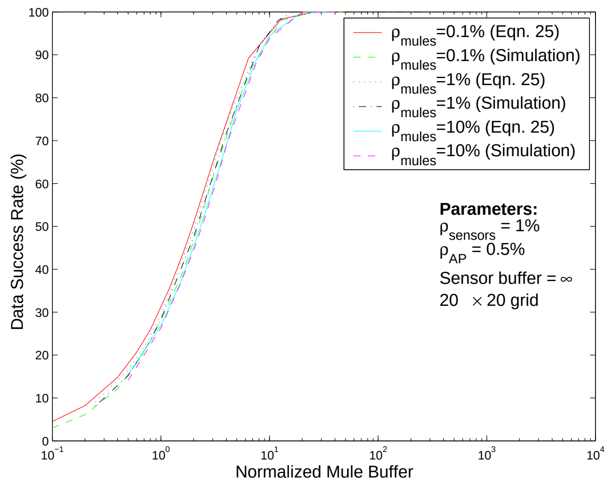

Fig. 10. Data success rate vs. normalized MULE buffer size for $\rho_{mules} = 0.1\%, 1\%$ and 10% (20×20 grid)

	ρ_{mules}	50% data success rate	90% data success rate
		MULE buffer	MULE buffer
sensor buffer $= \infty$	0.1%	$43,800$	$163,000$
	1%	4280	$14,800$
	10%	480	1660
		Sensor buffer	Sensor buffer
MULE buffer $= \infty$	0.1%	2030	7410
	1%	152	630
	10%	10	47

TABLE IV

SAMPLE VALUES OF MULE AND SENSOR BUFFER SIZES FOR 50% AND 90% DATA SUCCESS RATES

20×20 grid. The figure verifies that using the hitting time result for the continuized chain is valid for the discrete time case also. Similarly, Fig. 8 plots the cdf of $R_i^{N_{mules}}$ for a 20×20 grid with the same values of ρ_{mules}. Finally, Fig. 9 plots the cdf of R_{AP} for a mule on a 20×20 grid where $\rho_{AP} = 0.25\%, 1\%$ and 4%.

Figs. 10 and 11 plot the data success against the normalized MULE and sensor buffers respectively.

$$\text{Normalized MULE Buffer} = \frac{\text{Actual value of the MULE Buffer}}{E[\text{MULE Buffer}]} \qquad (26)$$

$$\text{Normalized Sensor Buffer} = \frac{\text{Actual value of the Sensor Buffer}}{E[\text{Sensor Buffer}]} \qquad (27)$$

For Fig. 10, the sensor buffer size was infinitely large. Note the steep drop-off of the data success rate with the MULE buffer size. Also, more than 95% data success rate is achieved when each MULE buffer is greater than $10E[M]$. Interestingly, the plot also shows that one can trade-off the number of MULEs in the system with the amount of buffer capacity on each MULE. This is evident from the fact that the data success rate curves are roughly the same for different MULE densities, but reducing the number of MULEs by a factor k increases the expected MULE buffer size by k (and vice versa). This will obviously impact latency, as the sensors will have to wait longer (or shorter as the case may be) before a MULE comes by to pick up the data. However, the analysis of the latency is left as future work.

Similarly, for Fig. 11 the MULE buffer size was infinitely large. Again, a steep curve was obtained for the data success rate. Also, the data success rate saturates for each MULE density when the sensor buffer capacity reaches roughly $10E[R_i^{N_{mules}}]$. However, the figure shows that we cannot trade-off a decrease in MULE density by increasing the buffers at each sensor. Higher MULE densities lead to higher data success rates, in general, until the sensor buffers are sufficiently large.

This can be seen more clearly in Table IV which shows the actual values of the buffer sizes needed to achieve data success rates of 50% and 90%. These are shown for both the cases of infinite MULE and infinite sensor buffers. For $SB = \infty$, the amount of MULE buffer needed to achieve a certain level of data success rate scales inversely as the mule density. However, when $MB = \infty$, the sensor buffer needs to increase (decrease) by a ratio greater than the decrease (increase) in the number of MULEs.

The reason for this can be understood by realizing what causes a drop in the data success rate. For the finite sensor buffer case, it is the sensor buffer getting full (thus, packets get dropped) that reduces the success rate; in the finite MULE buffer case, it is the MULE buffer overflowing that causes the reduction in success rate. Now, as the number of MULEs in the system decreases, the inter-arrival time at a sensor grows larger, and consequently, there are larger amounts of data that are dropped due to sensor buffer overflow. With increase in N_{mules}, the probability of no MULE arriving at the sensor for a long time reduces, hence the reduction in packet drops, increasing the data success rate. However, for the other case of infinite sensor buffer but finite MULE buffer, overflow of data on the MULE does not happen that often due to the Law of Large Numbers. Since the MULE picks up data from a large number of sensors, the probability of the total amount of data on the MULE exceeding the average value reduces. Thus the data success rate is not affected by packet drops when the number of MULEs change, as compared to the case when the sensor buffer is finite.

XI. Conclusion

The MULE architecture provides an economical method to connect sparse sensor networks at the cost of higher latencies. The main idea is to utilize the motion of the entities that are already present in an environment to provide a low power transport medium for sensor data. The data from the power-restricted sensors is carried by the MULEs and delivered to the access points. After introducing the architecture, the focus of the paper was on presenting a simple analytical model to provide an insight into various performance metrics (data success rate and buffer-sizes). The simple model enabled us to obtain analytical expressions for these metrics, which were then verified by simulations. Some of the key results from the analysis and the simulations were:

- The sensor buffer requirements are inversely proportional to ρ_{mules}.
- The MULE buffer requirement are inversely proportional to both ρ_{mules} and ρ_{AP}.
- When the sensor buffer is large the buffer capacity on each MULE can be traded-off with the number of MULEs to maintain the same data success rate.
- The change in the buffer capacity on each sensor needs to be greater than the change in the number of MULEs to keep the same data success rate.

An important issue that is not addressed in this paper due to lack of space is latency. Latency has two components - latency on the sensor before a MULE picks up the data sample and the latency on a MULE before it encounters an access point. We are currently working on obtaining distributions and bounds for both these quantities.

In addition, radio propagation and better mobility models such as Brownian motion with drift, Gauss-Markov model etc. need to be applied to get more realistic results. However, it is to be expected that with more complicated models, analysis will not be able to provide all the answers. From the protocol point of view, MULE-to-MULE communication and reliability using acknowledgments are other issues to be addressed in the future.

Acknowledgments

We would like to thank Jason Jenks (University of Washington) and Anthony La Marca (Intel Research, Seattle) for their active participation in defining the system architecture. Thanks are also due to Prof. Krzysztof Burdzy (Dept. of Mathematics, University of Washington, Seattle) and Prof. David Aldous (Dept. of Statistics, University of California, Berkeley) for providing us with information and references on non-interacting particle theory of Markov chains.

References

[1] D. Estrin et. al. Embedded, everywhere: A research agenda for networked systems of embedded computers. In *Computer Science and Telecommunications Board (CSTB) Report*, 2001.

[2] J. Rabaey et. al. Picoradio supports ad hoc ultra-low power wireless networking. In *IEEE Computer Mag.*, July 2000.

[3] J. Kahn, R. Katz, and K. Pister. Next century challenges: Mobile networking for smart dust. In *ACM/IEEE MobiCom*, 1999.

[4] UC Berkeley TinyOS. http://webs.cs.berkeley.edu/tos/.

[5] P. Juang et. al. Energy-effi cient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *Int'l. Conf. on Architectural support for programming languages and operating systems (ASPLOS)*, 2002.

[6] O. Dousse, P. Thiran, and M. Hasler. Connectivity in ad-hoc and hybrid networks. In *IEEE Infocom*, 2002.

[7] Amin Vahdat and David Becker. Epidemic routing for partially-connected ad hoc networks. In *Technical report, Duke university*, 2000.

[8] W. Brunette, D. Hoke, and J. Jenks. Mule. http://www.cs.washington.edu/education/courses/cse476/02wi/projectwebs/476mule/.

[9] D. G. Leeper. A long term view of short range wireless. In *IEEE Computer*, June 2001.

[10] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. In *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2002.

[11] Christian Bettstetter. Smooth is better than sharp: a random mobility model for simulation of wireless networks. In *Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 19–27, 2001.

[12] Z. Lei and C. Rose. Probability criterion based location tracking approach for mobility management of personal communications systems. In *Proc. IEEE GLOBECOM*, pages 977–981, 1997.

[13] Z. Lei and C. Rose. Wireless subscriber mobility management using adaptive individual location areas for pcs systems. In *Proc. IEEE International Conf. on Communications (ICC'98)*, pages 1390–1394, 1998.

[14] David J. Aldous and James A. Fill. Reversible markov chains and random walks on graphs. manuscript under preparation, http://stat-www.berkeley.edu/users/aldous/book.html.

[15] Robert Ellis. Torus hitting times from green's functions. http://www.math.ucsd.edu/ rellis/comb/torus/torus.html.

[16] Frank Spitzer. *Principles of random walk*. Springer-Verlag, 2001.

[17] Sheldon M. Ross. *Introduction to probability models*. Academic Press, 2000.