Exploiting Mobility for Energy Efficient Data Collection in Wireless Sensor Networks

Paper no. 540

Abstract— **There are two principal problems in collecting data in large and sparse ad-hoc sensor networks: 1) high energy expenditure in multi-hop routing between widely separated nodes; and 2) routing hotspots near the destination of the data that shortens the effective lifetime of the network.**

In this paper, we present and analyze an alternative architecture which addresses above problems. Our approach exploits mobile nodes present in the sensor field as forwarding agents. As a mobile node moves in close proximity to sensors, data is transferred to the mobile node for later depositing at the destination of the data. Transmitting data over these much shorter distances leads to substantial power savings at sensors. This is a natural scenario for a wide variety of monitoring and dissemination applications that span large geographic areas.

We investigate the advantages and disadvantages of our approach and present an analytical model to understand the key performance metrics such as data transfer, latency to the destination, and power. Parameters for our model include: sensor buffer size, data generation rate, radio characteristics, and mobility patterns of mobile nodes. The modeling results provide insights and guidelines for the deployment of such systems. Through simulation we verify that our approach can provide substantial savings in energy as compared to traditional ad-hoc network approach.

I. Introduction

Continuing advances in device and radio technology have enabled the production of small and inexpensive wireless sensor devices. These sensors will be embedded in the environment on a large scale and networked together to enable a wide variety of applications [1, 2]. Examples include: monitoring physical environments such as tracking animal migrations in remote-areas [3], weather conditions in national parks [4], habitat monitoring on remote islands [5], city traffic monitoring, seismic structure analysis, inventory tracking in warehouses, etc. For some applications such as city traffic monitoring or habitat monitoring, sensors are spread over a large geographic area resulting in a sparse network. The issues faced in efficiently collecting data in such large and sparse sensor networks are the focus of this paper.

The objective of monitoring systems is to collect data from sensors and deliver it to an access point to the infrastructure. These systems are expected to run *unattended* for long periods of time (on the order of months). The principal constraint is the energy budget of the sensors which is limited due to their size and needs to last as long as possible. In particular, the communication subsystem has been the primary energy consumption source [6, 2] and therefore solutions for energy efficient communication are of prime importance. In this paper, we present a solution for energy efficient data collection in sparse wireless sensor networks.

Current approaches involve forming an ad-hoc network among the sensor nodes to send data. However, this faces the following energy related issues. Firstly, in a sparse network, the energy required for transmitting data over one hop is quite large. This is because sensors may be far from each other and the transmission power required increases as the fourth power of distance. Secondly, in an ad-hoc network sensors have to not only send their data, but also forward data for other sensors. Thirdly, the network has routing hotspots near the access points. Sensors that are near the access points have to forward many more packets and drain their battery much more quickly. Finally, the routing protocol overhead in a large-scale network can be significant. This is true especially for protocols which try to optimize energy and typically require global knowledge [7]. In summary, we believe that there are significant challenges in using ad-hoc sensor networks for large-scale data collection in sparse configurations. Similar issues were encountered in the habitat monitoring project and an energy efficiency improvement of over an order of magnitude is desired to achieve the long terms of the project [5].

Our architecture addresses the above issues of per-hop energy, routing hotspots and routing overhead. The key idea is to exploit mobile entities present in an application scenario. We call these entities MULEs (Mobile Ubiquitous LAN Extensions) because they ”carry” data from sensor to access point. For example, in a city traffic monitoring application vehicles can act as MULEs; in a habitat monitoring scenario, the role can be served by animals; in a national park monitoring scenario, people can be MULEs. MULEs are assumed to be capable of short-range wireless communication and can exchange data as they pass by sensors and access points as a result of their motion. Thus MULEs pick up data from sensors, buffer it and later on drop off the data at an access-point. The resulting architecture can be viewed as having three tiers: sensors, MULEs, and access points. Of course, mobility can be applied to any combination of the three layers and they can be partially or completely collapsed (e.g., MULEs that also serve as access-points to the infrastructure).

In the MULE architecture sensors transmit data only over a short range that requires less transmission power. Further, sensors do not have to forward data for (as many) other sensors and there is little or no routing protocol overhead as a result. Therefore, substantial energy can be saved at the sensors. However, there are couple of limitations as well. Firstly, a sensor has to wait for a MULE to pass within range before it can transfer its data. Therefore, the observed latency in our architecture can be on the order of minutes or even hours. Nevertheless, for many applications such high latency is acceptable. For example, this is certainly the case for applications where data is collected for scientific analysis over a long time period. Secondly, in some sense the MULE architecture has transferred the burden of forwarding from sensors to MULEs. We expect MULEs to have much larger and more easily renewable energy resources than sensors. Whether the MULE approach provides a cost-effective solution or not is a difficult question to answer at this stage. We do not claim that the MULE architecture is always the method

of choice, but rather that for certain applications it may be the most effective option.

Although the MULE architecture is simple, significant issues have to be addressed to understand the performance-cost tradeoffs. We present a simple analytical model based upon queueing theory to understand the relationship between performance metrics and system parameters. Performance is characterized along three dimensions: data transfer rate, latency, and energy requirements at the sensors. Our model incorporates system parameters such as sensor data generation rate, buffer size, radio characteristics such as range and capacity, MULE velocity, MULE mobility model, etc.

The model allows us to answer questions such as how frequently MULEs should arrive at a sensor, how do buffer requirements at the sensors scale with MULE arrival frequency, how do radio characteristics affect data transfer, for what range of parameters is the queueing system stable etc.

We also use simulation to estimate the potential energy savings achieved with the MULE architecture as compared to forming an ad-hoc network. Our results are promising and indicate at least an order of magnitude energy savings (for communication). Energy savings increased were over two orders of magnitude for sparser networks. The potential improvement in the operational lifetime of the network was even more dramatic.

Another issue addressed is the efficient discovery of sensors. In the basic model, sensors continuously listen to discover nearby MULEs. We address this by lowering the sensor duty-cycle. Lowering duty cycle negatively affects performance and, based upon our analysis, we propose a novel discovery mechanism that permits significantly lower duty cycles while at the same time has very little impact on performance.

The paper is structured as follows. We next describe related work. Section III describes the model of our sensor network. We outline the limitations of existing approaches in section IV. The MULE architecture is discussed in Section V. Section VI describes the analytical model and derives various results. We evaluate our architecture in Section VIII. We discuss some enhancements in Section IX and conclude in Section X.

II. Related Work

We classify the related work in two parts. First we discuss the previous work that uses the concept of mobility for communication in ad-hoc networks. Then we briefly review existing work on energy efficient routing in ad-hoc networks.

A. Mobility for communication

Exploiting mobility for communication in ad-hoc networks has received much attention recently [8, 9, 10, 11, 12]. The work focuses on scenarios in which there is no immediate end-to-end path between two nodes that wish to communicate, usually because of limited radio range. If the nodes are mobile, end-to-end connectivity may be achieved by buffering data at the nodes and waiting to transfer until they are in range of access-points. Theoretical capacity of such networks was considered in [8]. It was shown that mobility can provide scalable throughput at the cost of latency. Controlling mobile nodes to achieve connectivity and efficiency has been discussed in [9, 12].

The general idea of our architecture is also *mobility*. The key difference is that our application context is focused on sensor networks unlike previous work where the focus was towards mobile ad-hoc networks. The severe resource constrained nature of sensors networks places different requirements on the optimization objectives. For example, our work tries to maximize sensor network lifetime by reducing the communication energy required at the sensors. This has not been discussed in previous work. Our architecture explicitly introduces a layer of mobile nodes for communication (MULEs) as part of the infrastructure.

More specifically, in the context of sensor networks, the ZebraNet [3] project collects data from sensors on zebras in a nature reserve by exploiting the natural motion of the animals. Therefore, sensors are themselves mobile and there are no explicit MULE(s). Mobile access-points, in the form of overflying aircraft, have also been suggested. Our architecture also targets fixed sensor networks and encompasses the ZebraNet scenario. The Manatee project [13] is also exploring the idea of using mobility. A weather monitoring application in a national park is discussed in [4]. There are three distinctions with our work. Firstly, we derive an analytical model for understanding performance metrics. Secondly, we show the trade-offs between our approach and traditional approaches using ad-hoc networks in the context of data collection. Finally, energy efficient operation of sensors, such as discovery, which is central to our discussion has not been addressed in [4].

Another related project is the Infostation project which provides services such as email/file-transfer to a mobile user [14]. *Infostations* are installed throughout the city and act as very high bandwidth data exchange points. A mobile user can fetch the data required whenever they are in the vicinity of an Infostation. Our domain of sensor networks presents different constraints and objectives, particularly as regards to power, as compared to the Infostation project.

B. Energy Efficiency in Ad-hoc Networks

Energy efficient routing in ad-hoc networks has been addressed in [7, 15, 16, 17, 18, 19]. Inspite of the plethora of work, extending these ideas efficiently to large scale sensor networks remains a challenge [20, 5, 21]. For ad-hoc networks optimizing overall energy and maximizing network lifetime are different goals and a solution for one does not transfer to the other [7]. Optimizing for network life is much harder as it requires use of multi-path routing to eliminate hotspots [16, 22, 23]. Our architecture on the other hand results in equal consumption of energy at sensors because they are not responsible for forwarding and therefore experience no hotspots.

Another class of techniques optimize energy by reducing the radio listening time. For dense networks this is achieved by making a subset of nodes go to sleep [17, 19]. For sparse networks, techniques based upon reducing the duty-cycle are proposed [24]. We also reduce a sensor's duty-cycle to minimize the radio listening overhead.

III. Model and Metrics

We now describe the class of sensor networks for which our architecture is suitable. We also discuss performance metrics for

gauging the effectiveness of a data collection solution in such a sensor network.

Model

Our architecture is designed for large and sparse wireless sensor networks where mobile entities are present. Sparse networks occur when there are relatively few sensors covering a large geographic area such as a city or forest. Sensors are assumed to be small, resource (energy, memory, bandwidth, CPU) constrained and battery operated. To achieve increased longevity of the network it is crucial to efficiently utilize resources available at sensors.

The purpose of a sensor network is to sense the environment and transfer the sensed data to the infrastructure for further elaboration. Our architecture is targeted for applications in which the data is sent to external storage through a small number of *access-points*, which are servers with ample storage and Internet connectivity. Access-points can communicate to each other and therefore, in our model, it is sufficient that data reaches any one access-point. Furthermore, the MULE architecture is applicable only when real time delivery of data is not required. This is true for many monitoring applications where data is collected for scientific analysis over time.

Example Scenario: Traffic Monitoring

Sensors are scattered over various street intersections in a city to collect a variety of data including: counting the number of vehicles, vehicle speeds, traffic density, etc. The data is transmitted to a central repository where it can be stored for future use by traffic engineers to determine optimal timings of traffic lights or the need for additional lanes. In this scenario, the MULEs can be vehicles such as mail-vans, police cars, buses, and/or taxis. These vehicles can be fitted with an appropriate transceiver to discover and collect the data from the sensors as they drive around. Later on, as they are in the vicinity of an access-point the data is delivered to a central repository via the internet.

Metrics

- **Data Success Ratio (DSR)**: This measures the effectiveness of data delivery. It is defined as the ratio of the total amount of data transferred to the access-points to the total amount of data generated. This metric has been also been used in [3, 25]. Ideally, DSR will be one. Data may be lost because of errors in radio communication or failure of MULEs. In addition, the sensor's limited buffer capacity (for example, a UC Berkeley MICA mote has only 500KB) may also cause data loss. If no MULE comes for a long period of time the buffer may fill and, eventually, overflow. Average sensor buffer occupancy is used as an indicator of sensor buffer requirements.
- **Latency**: This is the average time taken by data to reach access-points from the time of its generation. The latency requirements are application dependent. For example, in a traffic monitoring application latency could be many minutes or even hours. Interestingly, latency will be lower in cases of higher traffic corridors where there are more vehicles and the need for more data at lower latency.
- **Communication Energy**: Modeling complete energy consumption is a complex subject in itself; therefore, in this paper we will focus on the energy required for communication. We assume (as argued by others [2]) that by saving energy required for communication, lifetime can be improved substantially. Within this context the following metrics are evaluated [26]:

Average Usage: Average energy consumed per sensor in communicating data from sensors to access-points.
Hotspot Usage: Maximum energy consumed by any sensor. This dictates the network life time because this determines the time till one of the sensor runs out of energy.

IV. Current Approaches

In this section, we discuss the limitations of current approaches for data gathering.

A. Ad-hoc Network

In ad hoc networking approaches, the sensor nodes form an ad-hoc network to gather data [6, 27, 5]. It has the following limitations:
1. **Large Transmission Power:** In a sparse ad-hoc network the average distance between two neighbor sensors is large. The power required to transmit packet increases dramatically with distance (typically as the fourth power of distance [28]); therefore, a substantial amount of energy is consumed to send a packet over a single hop. In addition, a packet typically traverses multiple hops before reaching its destination, an unnecessary waste of communication energy.
2. **Routing Hotspots:** Sensors located near access-points forward many more packets than others and consequently form hotspots. This reduces the network lifetime substantially. We verify and discuss this further in the evaluation of section VIII-C.
3. **Routing Scalability:** Current ad-hoc routing algorithms that optimize energy consumption or network lifetime are based on a global knowledge [7, 16, 19, 21]. Techniques based upon hierarchy [23] or location information [29] have been proposed to reduce the overhead by a limited extent but are also more complex. We believe that the routing overhead; *computation, memory and energy*; of current energy-aware routing protocols for networks containing thousands of sensors is substantial and a limiting factor in large-scale sensor deployments.
4. **Extra Forwarding Nodes:** If the sensor network is sparse and the radio range is small, it is possible that the ad-hoc network is not fully connected [30]. Extra forwarding nodes are required to make the network dense enough to achieve connectivity.
5. **Radio Listening Overhead:** In a basic ad-hoc network, sensors have to continuously listen because they may have to forward data for other sensors. Listening consumes substantial energy and reducing this overhead is important [19, 24, 31, 32]. One approach is to reduce sensors' duty-cycle and listen only when required. This is challenging because in ad-hoc networks there is no centralized entity and decisions about when to listen have to be taken in a distributed and cooperative fashion [32].

B. Direct Communication

In this approach, each sensor transmits data directly to one of the few access-points. This leads to large energy consumption as typically the nearest access-point is located far away from the

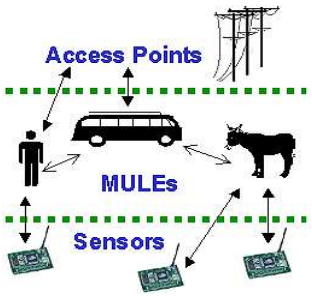


Fig. 1. The three tiers of the MULE architecture

sensor. Deployments of access-points are usually limited because of their cost. An access-point is equivalent to a server and has similar installation and recurring maintenance costs. Moreover, in some environments installing multiple access-points may not even be feasible because of physical limitations such as in remote-terrains.

V. MULE Architecture

This section describes our basic MULE architecture. Enhancements to the basic architecture are described in section IX.

A. Overview

The MULE architecture provides connectivity by adding an intermediate layer of mobile nodes to the existing relationship between sensors and access-points used in typical sensor network designs [5, 26] as shown in Figure 1. As a MULE moves in close proximity to a sensor, the sensor's data is transferred to the MULE for later delivery to an access-point. Transmitting data over these short distances leads to substantial power savings for sensors. However, a sensor has to wait for a MULE to pass nearby before it can send data which makes the latencies much higher. We now describe the three-tiers and the interactions among them.

B. Three-Tiered Design

- **Lower tier - sensors**: Sensors provide data, communicate via a short-range radio, and have limited power and memory. The amount of work performed by sensors should be minimized because they have the most constrained resources among the three tiers. In our architecture, sensors' communications are limited to transferring data to a nearby MULE.
- **Middle tier - MULEs**: MULEs are mobile entities with large storage capacities (relative to sensors), renewable power, and have the ability to communicate with sensors and access-points. A MULE has the responsibility to discover sensors and access-points and transferring data between them. In our basic model MULE(s) do not communicate with each other. In Section IX we discuss the effect of MULE-to-MULE communication as an enhancement to our basic architecture.
- **Upper tier - access-points**: These are servers with Internet connectivity and enhanced power, storage and processing capabilities. For our purposes, these are the eventual destination of sensor data. They are used to offload the data collected by and stored in the MULEs.

The MULE system is intended to create a framework that can be applied to a large variety of sensing applications. The MULE layer is an abstract network layer for the endpoints (sensors and access-points) and can be used simultaneously by different applications.

Depending on the scenario, a number of tiers in our three-tier abstraction could be collapsed onto one device. This increases the applicability of our architecture. For example, sensors can be mobile as in the ZebraNet project [3]. Here sensors are attached to zebras, causing the sensor and the MULE tier to be mapped to the same device. Similarly, if MULE(s) have Internet connectivity and sufficient storage they can act as an access-point, thus, combining the MULE and access-point tiers. For example, to reduce latency in the traffic monitoring application, MULEs can be equipped with an always-on connection (such as a cellular modem) that allows the MULE to transmit sensor data immediately to an access-point and thereby reducing the latency between the upper and middle tiers. For example, this would be appropriate in making traffic monitoring more real-time.

C. Data Transfer Interactions

- **Discovery**: A sensor needs to discover a nearby MULE to be able to offload its data. In our architecture the prime responsibility of discovery is placed on the MULE, as our objective is to minimize the load on sensors. A MULE continuously sends out a discovery message to detect a nearby sensor. This requires a sensor to listen for discovery messages. For most radio technologies listening can consume a substantial amount of power, almost as much as receiving [17]. Therefore it is important to reduce the amount of time a sensor spends listening. As discussed earlier in Section IV-A, the same issue exists in ad-hoc networks, where a node has to listen continuously because it might be required to forward data for other node.

  The situation is simpler in the MULE architecture because a sensor is not responsible for forwarding another sensor's data and can make decisions about listening locally. To save additional power, we apply the basic idea of reducing the duty cycle of the sensor radio. A sensor can periodically (at a low rate) listen to the radio channel to discover nearby MULEs. Clearly, there is a tradeoff as a sensor may miss some MULEs and performance may deteriorate. One can have very low duty cycle where the radio listening overhead is lower but performance is worse because sensors don't discover MULEs as much. This affect of duty cycle on performance is analyzed in detail in Section VII-E. We also discuss some interesting techniques for reducing listening time in the enhancements section IX.
- **Data transfer**: In our basic model, the sensor transmits as much data as it can to the MULEs in the order the data was generated. In Section VII-D we will derive an estimate of how much data can be transferred as a MULE passes by a sensor. However, the amount data that needs to be transfer between MULEs and access-points is much larger, as MULEs carry data for multiple sensors. To solve the problem a high bandwidth radio may be used for MULE to access point communication, or the MULEs can increase the amount of time they are near an access point. For example in the traffic monitoring application, mail trucks (acting as MULEs) park every night at a post office, giving plenty of time for their data to be offloaded.

D. Trade-offs

We now highlight the relative advantages and disadvantages of the MULE architecture.

Benefits

- **Energy Efficient:** Substantial energy is saved because sensors communicate over a short range. Moreover, there are no hotspots in the network as sensors do not forward data for other sensors. Energy savings are evaluated in Section VIII-C.
- **No Routing Overhead:** In contrast to ad-hoc networks, the MULE architecture does not have any routing protocol overhead for sensors.
- **Robustness:** Performance degrades gracefully as MULEs fail. Any single MULE failure does not lead to a disconnected network. The primary effect of a MULE failure on the overall system is a slight increase in latency as there are now fewer MULEs to pick up data. In contrast, in an ad-hoc network failure of few critical nodes might lead to a disconnected network.
- **Scalable:** The MULE architecture is easily scalable as deployment of new sensors or MULEs requires no network reconfiguration.
- **Simplicity:** The data routing aspect of the MULE architecture is very simple and extremely lightweight for the sensors. This is important because sensors are the bottleneck of the system. The MULE architecture does not require any synchronization or location information; an assumption made by many ad-hoc networks based solutions [26, 19]. The MULE architecture also exploits spatial reuse of bandwidth by using short-range communication without losing long term connectivity and avoids radio communication complexities such as collisions.

Limitations

- The MULE architecture has high latency and this limits its applicability to real-time applications (although this can be mitigated by collapsing the MULE and access-point tiers).
- The system requires a sufficient number of mobile nodes in the application environment to act as MULEs (often this scales appropriately with the application as is likely to be the case for traffic monitoring - more traffic leading to naturally more MULEs and more timely data collection).
- Data delivery in the basic architecture is best-effort; delivery is not guaranteed. There are two reasons for this. First, MULEs motion may be quite random. They may not arrive at a sensor or after picking the data may not reach near an access-point to deliver it. Second, data may be lost because of radio-communication errors or MULEs crashing. To improve data delivery, higher-level protocols need to be incorporated in the MULE architecture. This is discussed further in the enhancements section IX.

VI. Analytical Model

The goal of our modeling is to understand the relationship between performance metrics and parameters in the MULE architecture. As discussed in section III the metrics are: average sensor buffer occupancy, DSR (the fraction of generated data that is delivered to the access-points), latency. Modeling energy requirements for communication is considered later in Section VIII-C.

We begin with a discussion of the parameters involved in the MULE architecture. .This is followed by an analytical model based upon queuing theory and the results for the different performance metrics.

A. Parameter Space

The parameter space can be divided into the four following categories.

- **Sensor related**: The data generation rate (λ) defines the average amount of data that a sensor is generating. This directly affects the buffer requirements at the sensor. The sensor buffer size (SB) determines the maximum amount of data that can be stored on the sensor and can affect loss of data from buffer overflows. Another parameter is the duty cycle of sensor.
- **MULEs related**: The primary aspect is to determine when MULEs come into the communication range of a sensor. The MULE arrival within a sensor's range is modeled as a discrete event. Thus, the key parameter is the distribution of time between two MULE arrivals at a sensor.

Determining this distribution is a complex problem that depends on many factors such as MULE velocity, number of MULEs, sensor's radio range and a MULE's mobility pattern. For example, doubling the number of MULEs or doubling velocity would double the average MULE arrival rate. The distribution of arrivals also depends on the application scenario. In a traffic monitoring scenario, if the MULEs are city buses then the inter-arrival distribution can be modeled as deterministic; whereas, in a habitat monitoring application the MULEs are animals, the inter-arrival distribution would be determined by the random motion of the animals.

MULEs buffer size is another parameter, but for the purposes of this paper we assume that MULEs have sufficiently large buffers.

- **Access point related**: The important aspect here is the distribution and the number of access-points. This affects how frequently MULE visits access-point to deliver data. This is modeled by a parameter characterizing the distribution of the time interval between visits to access-point by a MULE.
- **Radio related**: The radio parameters affect the amount of data that can be transferred as a MULE passes by a sensor. We use a radial model for the radio, i.e. sensors and MULEs can communicate if they are within a distance r. The rate of data transfer is a fixed quantity B. Although simplistic, this provides a good approximation, particularly because the sensor to MULE communication will be over a short-range.

The discussion of the categories above highlights the fact that there are many knobs in the MULE architecture. Our approach is to identify a few basic parameters that are sufficient to characterize the performance metrics. These basic parameters are: 1) sensor data generation, 2) sensor buffer size (SB), 3) amount of data transferred between a MULE and a sensor, denoted by K 4) MULEs arrival at a sensor and 5) a MULE's visit to access-points. These are defined more precisely in the next section.

The affect of other parameters can be understood by first studying how they change one or more of the basic parameters and subsequently studying how the performance is affected by the change in basic parameters. For example, the impact of increasing MULE velocity on performance can be examined in two steps. First, by examining the impact of increasing MULE velocity on the basic parameters. In this case, it increases the

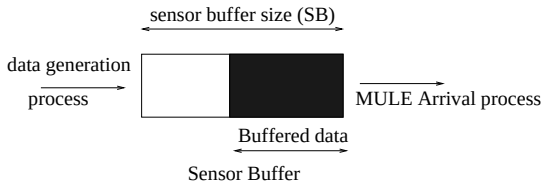


Fig. 2. The queue model for MULE architecture

MULE arrival rate at the sensors/access-points and decreases K (see section VII-D). Second, the analytical model is used to analyze the affect on performance due to the changes in these basic parameters. The effects of sensor duty cycle are modeled in a similar manner (see section VII-E).

B. Model

The primary component of our model is a queue of generated data (but not delivered) at each sensor. In queuing theory terminology, generation of new data at a sensor corresponds to an arrival at the sensor's queue. The buffer size of the sensor defines the capacity of the queue. If the buffer is full then any newly generated data is dropped. The queue is served whenever a MULE is in a sensor's range. For modeling purpose the arrival of a MULE in a sensor's range is considered as a discrete event. This event causes transfer of data from the sensor's queue to the MULE. The sensor then waits for the next MULE arrival event to transfer the data. Thus, the time between two MULE arrivals defines the service time.

The amount of data that be transferred on a MULE arrival event is a random variable and depends on factors such as, the time the MULE is in the communication range of sensor. However, for analytical tractability, this is taken as a fixed quantity, denoted by K. K is derived in section VII-D.

The above model assumes that the MULE(s) can transfer all the data to the access-points. We do not model in detail the interactions between the MULEs and the access-points. Our focus is on sensors, which are the primary bottleneck of the system. In future, we plan to extend our analysis to incorporate MULE-access-points interactions in detail.

The above queueing model resembles the bulk service model in the queuing literature. The model is typically denoted as $G/G^K/1/SB$ [33]. The two G's stands for the general input (data generation) and service (MULE arrival) distributions respectively. K is the service size, and SB is the maximum queue capacity. If less than K units of data are available at the sensor then that data is transferred and the MULE leaves without waiting for additional data.

The following list provides a summary of assumptions and key notational symbols. For a complete glossary refer to appendix -A.

- The MULEs arrival process at a sensor is a renewal process $\{S(t), t \geq 0\}$, where $S(t)$ is the total number of MULEs that have visited the sensor till time t. The renewal assumption means that the inter-arrival times (time between arrival of two MULEs) are independent and identically distributed (denoted by random variable X^s). Average MULE arrival rate is denoted by μ and the variance of X^s is σ_{ms}.
- A MULE's visit to the access-points is also a renewal process $\{R(t), t \geq 0\}$, i.e, the time intervals between a MULE's visit to access-points are independent and identically distributed. Average MULE visit rate is denoted by μ_r and the variance of the inter-arrival times by σ_{mr}.
- At a given time only one MULE interacts with a given sensor and vice-versa.
- Sensors are identical. This allows us to generalize by analyzing any one sensor. Although not essential, we will assume that sensors are not mobile for ease of exposition.
- The data generation process at a sensor is a renewal process $\{U(t), t \geq 0\}$, where $U(t)$ is the total amount of data generated till time t. Average data generation rate is denoted by λ.
- The queueing discipline is FCFS. The data that is generated first is picked up first.
- MULEs have sufficiently large buffers.
- A MULE is able to transfer all its data to an access-point whenever they come in contact.
- Without loss of generality, $SB \geq K$. If $SB < K$ then the maximum amount of data that is available at sensor buffer to transfer to MULE is SB. Therefore, $K = SB$ for such cases.
- Data transmission does not incur any loss. The only loss is due to sensor buffer overflow.
- The queueing system is stable and only the stationary (time independent) probabilities are considered. These are the probabilities as $t \to \infty$.

VII. Results

A. Stability Condition

***Result** 1:* The system is stable (the queue reaches a unique stationary regime) iff

$$\frac{\lambda}{K\mu} \leq 1 \tag{1}$$

Proof: The proof is given in [34] (Theorem 3.1). ■

Intuitively, the equation says that the system is stable if the net service rate (product of K and the MULE arrival rate) is more than the data generation rate. The utility of this lies in the fact that if the stability condition is not satisfied, the sensor queues can grow arbitrarily large leading to data loss and large latencies.

Our analysis assumes that $SB \geq K$ (see assumptions VI-B). Incorporating this we get,

$$\frac{\lambda}{min(SB, K)\mu} \leq 1 \tag{2}$$

The above equation can be used to derive the minimum value of K or SB (for a given λ, μ) required to reach a stable system.

B. Results for Performance metrics

We now present results for different performance metrics. The rest of this section assumes the knowledge of the distribution of the queue length at the instance a MULE arrives at a sensor (denoted by the random variable Q). More specifically, P_j will denote the probability that the queue length Q is j (note that $P_j = 0$ for $j > SB$). Distribution of Q for specific scenarios is derived in next section.

The average of Q ($E[Q]$) is used as a measure of the average buffer occupancy of a sensor. By definition,

$$E[Q] = \sum_{j=0}^{SB} jP_j$$

$E[Q]$ indicates the sensor buffer requirement. In general, the sensor buffer (SB) should be much larger than $E[Q]$ to prevent any loss of data. From the definition of Q, this quantity is also the average amount of data that a sensor will have when a MULE comes nearby. Therefore, $E[Q]$ can be used as an indicator of K also.

***Result** 2:* Data Success Ratio (DSR) is given by:

$$DSR = \frac{\mu E[min(K,Q)]}{\lambda} \quad (3)$$

$$= \frac{\mu(\sum_{j=0}^{K} jP_j + \sum_{j=K+1}^{SB} KP_j)}{\lambda} \quad (4)$$

Proof: Proof is given in appendix -C. ■

Later, we will see that $P_j's$ depend only on the ratio of λ and μ and not on their absolute values. From the above equation this will also be true for DSR. This tells us that the system performance (DSR and buffer occupancy) will not be affected if both parameters are scaled proportionately.

The rest of this section deals with the derivation of average latency. Latency has two components. The first component is the queuing delay which is the amount of time spent by data in the sensor queue (W^q). The second is the time spent by data on a MULE before it is delivered to an access-point (W^m).

***Result** 3:* Average queuing delay (W^q) is given by:

$$W^q = \frac{\mu^2\sigma_{ms} + 1}{2\mu} + \frac{E[B^{no}]}{\mu} \quad (5)$$

Proof: In general, a single MULE may not be able to transfer all the data in the sensor buffer. In such a case multiple MULEs may have to arrive before a data sample is served. $E[B^{no}]$ denotes the average number of MULEs that arrive at the sensor while a data unit is in the queue excluding the MULE which serves the data unit itself. The expression for $E[B^{no}]$ is derived in appendix -D. Recall that, for the MULE arrival process $\{S(t)\}$, μ is the average renewal rate and σ_{ms} is the variance of the inter-arrival time distribution.

Consider a random time t at which some data (call it d) is generated and accepted into the queue. The time spent by d in the queue can be decomposed into two parts. The time till the next MULE arrives after t, plus, the time till next $E[B^{no}]$ MULEs arrive. This is because on average d is served when the $(E[B^{no}] + 1)$'th MULE arrives.

To compute the average time till the next MULE arrives, we will use the concept of Residual Life for renewal processes. This is a standard concept in the theory of renewal processes and for completeness sake is briefly discussed in Appendix -B.

Since the MULE arrival process is a renewal process, the average time till the next MULE arrival is by definition the average residual life of the MULE arrival process ($\{S(t)\}$). Therefore, by residual life theorem -B.1, the average residual life for $\{S(t)\}$ is: $\frac{\mu^2\sigma_{ms}+1}{2\mu}$.

Since the average time between two arrivals of MULE is $\frac{1}{\mu}$, the average time taken for $E[B^{no}]$ MULEs to arrive is $\frac{E[B^{no}]}{\mu}$.

Finally, W^q is the sum of the above two components. ■

If K is sufficiently large, a MULE can pick up all the data in the sensor queue. In this case $E[B^{no}]$ would be zero. Therefore, the average queuing delay is just the residual life of the MULE arrival process. The average queuing delay increases with σ_{ms}. Therefore, MULE arrival processes with lower variance will have lower queuing delay.

***Result** 4:* Average time spent by a packet on MULE (W^m) is:

$$W^m = \frac{\mu_r^2\sigma_{mr} + 1}{2\mu_r} \quad (6)$$

Proof: Recall that for $\{R(t)\}$, μ_r is the average renewal rate and σ_{mr} is the variance of the inter-arrival time distribution. The proof is similar to the derivation of the first part of the queuing delay, W^q.

Consider a random time t at which some data (call it d) is transferred to the MULE. The amount of time d spends on the MULE is the time from t until the MULE visits the next access-point. Since MULE's visits to the access-points is the renewal process $\{R(t)\}$, this is the average residual life of the process $\{R(t)\}$. By residual life theorem -B.1, the average residual life for $\{R(t)\}$ is $\frac{\mu_r^2\sigma_{mr}+1}{2\mu_r}$. ■

Additionally, when $\{R(t)\}$ is a poisson process, the average residual life is simply $\frac{1}{\mu_r}$ (use corollary -B.1.1 in appendix -B). Therefore in this case W^m can be simplified to:

$$W^m = {\mu_r}^{-1} \quad (7)$$

Lemma VII-B.1: Average latency (W) is:

$$W = W^m + W^q \quad (8)$$

This follows directly from the observation that the latency seen by a data is the sum of queuing delay and the time spent on the MULE.

C. Specific scenarios

The previous section has presented results assuming that the distribution of Q is known. We now derive Q for specific scenarios.

C.1 The MULE arrival distribution and the data generation process is Poisson

The poisson assumption allows us to obtain closed form results. Moreover, it can be a reasonable approximation under certain environments. For example, it is known by the Palm-Khintchine theorem (p. 156 [35]) that under mild conditions on the individual arriving entities (MULEs in our case), the aggregate arrival process (also called the superposition process) often looks approximately Poisson as $n \to \infty$.

We directly apply the results from Section 4.5 of [33].

$$P_j = (F_{SB-j} - F_{SB-j-1})/[F_{SB}], j = 0, \ldots SB-1$$
$$P_{SB} = 1/[F_{SB}]$$

where,

$$F_0 = 1$$
$$F_i = \sum_{s=0}^{[i/(K+1)]} (-1)^s \binom{i-sK}{s} (1-p)^s p^{sK-i} \quad i \geq 1$$

Observe that P_j's depend only on the ratio of μ and λ. This indicates that the absolute value of μ and λ is not important. This would be useful in evaluating the effect of scaling parameters on performance (see section VIII) as one of the parameters can be fixed.

C.2 K is large ($K \geq SB$)

When $K \geq SB$, all the data is transferred when a MULE visits a sensor. Therefore, the amount of data in the sensor buffer (Q) is the minimum of: 1) the amount of data generated during the time between arrival of two MULEs, 2) the sensor buffer size. In most cases, by stationarity assumption, the amount of data generated in an interval depends only on the length of the interval. For example, for poisson or deterministic data generation process. Therefore,

$$Q = min(U(X^s), SB)$$

If SB is large, the equation can be further simplified to:

$$Q = U(X^s)$$

In this case, the expected queue length can be derived simply as:

$$E[Q] = E[U[E[X^s]] = \frac{\lambda}{\mu} \tag{9}$$

D. Determining K

K is the average amount of data that can be transferred between a MULE and a sensor, as the MULE passes by a sensor. We assume that the sensor is stationary. For mobile sensors the same analysis can be applied by considering the relative motion between the two entities.

In our radio model, sensors and MULEs can communicate only if they are within a distance r. Therefore, the amount of data transferred is the radio data transfer rate (B) times the amount of time the MULE is in the radio range of sensor (called CT)[1].

$$K = CT \times B \tag{10}$$

The average contact time can be computed as follows. Let x be the perpendicular distance between the sensor and the MULE's line of motion as shown in Figure 3[2]. Assume that x is uniformly distributed between 0 and r. If x is greater than r then the MULE is not in contact with the sensor. The average distance that the MULE remains in contact with the sensor can now be computed as:

$$2 \int_{x=0}^{r} \frac{\sqrt{r^2 - x^2}}{r} dx$$

[1]We are ignoring the time required for discovery. This is reasonable because typically the discovery time would be much smaller than the time the MULE will be in contact with the sensor.

[2]In general an application may have additional constraints on x, such as for traffic monitoring application x is at-least few meters because of spatial constraints.

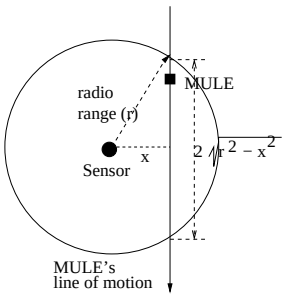


Fig. 3. Illustrates the amount of time a sensor is in contact with a MULE

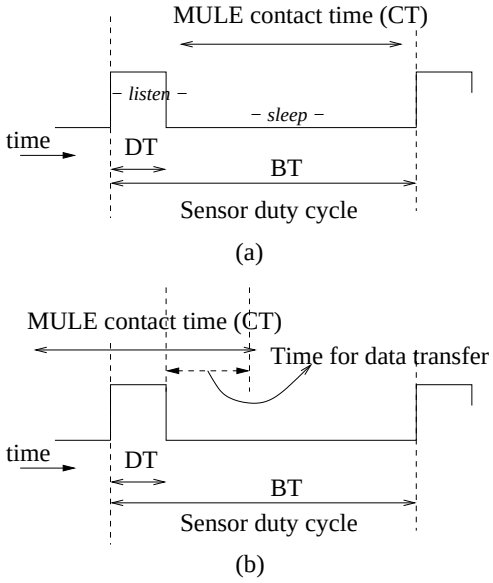


Fig. 4. Illustrates the impact of duty cycle on discovering a MULE. (a) the MULE is missed by the sensor because the sensor is asleep during the time the MULE is in contact with the sensor. (b) The sensor discovers the MULE after it has been in it's range for some time. This leaves only a fraction of contact time for data transfer.

The above integral evaluates to $\frac{\pi}{2}r$. If the MULE has a velocity v, we get

$$CT = \frac{\pi}{2}\frac{r}{v} \tag{11}$$

Combining equation 10 and 11,

$$K = (\frac{\pi}{2}\frac{r}{v})B$$

For example, consider a sensor-MULE interaction using a Berkeley mote. The Berkeley mote has a radio range of about 25 meter and data transfer rate of 40Kb per second. If the MULE has a velocity of 10 m/s (10 m/s is approx 20 miles per hour), using above equation, we get $K = 150Kb$.

E. Impact of sensor duty cycle

We will assume that the sensor periodically listens for DT seconds every BT seconds. DT is the time required for the discovery protocol to complete and BT is the beacon interval time. Duty-cycle (γ) by definition is, $\frac{DT}{BT}$. Compared to the 100% duty cycle case, performance will be affected because of two reasons:

1. A MULE may not be discovered at all because the sensor was asleep during the time the MULE was in communication range of sensor (Figure 4a). This affects the effective MULE arrival rate, which is the rate at which a sensor actually discovers a MULE. We model this by finding the probability of discovering a nearby MULE and use it to get the effective MULE arrival rate. For example, if the probability of discovering a MULE is 0.25, then the effective arrival rate is one-quarter of the original rate[3].
2. The amount of data that can be transferred(K) may decrease if the MULE is not discovered in the beginning, but in the middle of the time it is in the communication range of the sensor. We model this by finding an effective K, the average amount of data transferred between the MULE and the sensor due to late discovery.

We now derive the effective MULE arrival rate (called μ^*) and effective data transfer (called K^*). CT as before, denotes the time the MULE is in the communication range of sensor.

Effective MULE arrival rate

If $CT \geq BT+DT$, then the sensor is awake for the discovery time at least once during the time the MULE is in range of the sensor [4]. Therefore if $CT \geq BT$, the probability of discovery is 1 and the effective arrival rate remains unchanged.

However, if $CT < BT+DT$, there is a window during which a MULE may be missed. This is illustrated in Figure 4 (a). The probability that a MULE is missed is the same as the probability that the MULE contact time interval does not overlap with the sensor's discovery interval. Assuming that the MULE contact time can begin uniformly at any time with respect to a sensor's duty cycle, the probability of discovering a MULE is $(CT - DT)/BT$. Therefore, μ^* is $\mu(CT - DT)/BT$.

Effective data transfer

If $CT \geq BT+DT$, then as discussed above the sensor is discovered with probability one. In fact, the discovery starts in the first BT seconds. Assuming that the discovery is equally likely to begin at any time during the first BT seconds, the average time before discovery starts is $BT/2$. Once the discovery starts, the sensor is in contact with the MULE for the rest of the contact time period. Hence, the average contact time is $CT - (BT/2)$. Therefore, K^* is $K(1 - BT/2CT)$.

When $CT < BT + DT$, if the MULE discovers the sensor then the discovery starts in the first $CT - DT$ seconds. Therefore, the average elapsed time before discovery starts is $(CT - DT)/2$. Thus, the total average contact time is $CT - (CT - DT)/2$, which is $(CT + DT)/2$. Therefore, K^* is $\frac{K}{2}(1 + DT/CT)$.

Summary

We now summarize the effect of a low duty cycle for the interesting case when the duty cycle is very low (BT is large). Specifically when $BT > CT$.

$$\mu^* = \mu\gamma\frac{CT}{DT} \quad (12)$$

$$K^* = \frac{K}{2} \quad (13)$$

The following observations can be made from the above two equations:
- The effective MULE arrival rate is proportional to the duty cycle γ. This is expected as lowering the sensor duty cycle decreases the discovery probability and hence reduces the effective MULE arrival rate.
- The effective amount of data transferred in a single contact is roughly halved and is independent of the duty cycle.
- The decrease in the effective MULE arrival rate because of low duty cycle can be compensated in two ways. One method is to decrease the discovery time (DT). Reducing DT will also reduce the duty cycle. Therefore, low latency discovery protocols are important. The second method is to increase the contact time CT. As discussed in the previous section, CT depends on the radio range and the MULE velocity and therefore can be controlled to a certain extent. An interesting enhancement for efficient discovery, which exploits this method is discussed in more detail in Section IX.

As an example, suppose sensors have a duty cycle of $1/100$. Consider a sensor-MULE interaction scenario, where the radio range is 25m and the MULE velocity is 10 m/s. The contact time (CT) for these parameters is approximately 4 seconds using Equation 11. Discovery time is typically 10's of milli-seconds, say 40 ms. For these parameters, from equation 12, μ^* is the same as μ. The only affect is on K, which is halved. This shows that the sensors can operate at low duty cycles without substantially affecting performance.

VIII. Evaluation

This section evaluates the following aspect of the MULE architecture using both analysis and simulation.
1. **Performance Metrics:** This section investigates the effect on performance metrics as system parameters are scaled. The results derived in the analytical modeling section VII-B and VII-C.1 (when the MULE arrival process is Poisson) are applied here.
2. **Mobility Model:** The impact of the MULE mobility model on performance is studied. Other than Poisson arrivals, MULE arrival processes governed by mobility models such as deterministic, random-waypoint and manhattan [37] are discussed.
3. **Energy Savings:** This quantifies the energy savings and the increase in the lifetime in the MULE architecture as compared to an ad-hoc network.

A. Performance Metrics

This section investigates the effect on performance metrics as system parameters are scaled. The parameters and the performance metrics considered are the one's defined in the modeling section VI.

The data generation process and the MULE arrival process are assumed to be poisson and apply the results presented in Section VII-B and Section VII-C.1. Data generation rate (λ)

[3]Here we are assuming that the MULE arrival process is Poisson and the results hold because random sampling of Poisson processes results in another Poisson process [36]. For general distributions, this provides a convenient approximation.

[4]The sensor has to be awake for a consecutive time period of DT seconds. That's why the condition is $CT \geq BT + DT$ instead of $CT \geq BT$

is fixed at 90KB/Hour. This is reasonable because as mentioned during analysis the absolute value of λ is not important and only the ratio $\frac{\lambda}{\mu}$ affects the performance metrics. Therefore, it is sufficient to only scale μ (the MULE arrival rate at a sensor). Also, to simplify presentation the time spent by data on a MULE is not considered. Under poisson assumption, this is the inverse of the rate at which the MULEs visit access-points (Equation 7) and is trivial to incorporate.

We first study the effect of increasing μ and SB, assuming sufficiently large K ($K \geq SB$). Subsequently, the effect of K is considered.

A.1 Scaling μ and SB

Figure 5 shows the effect of increasing μ on the performance metrics. The three different lines on the plots corresponds to three different sensor buffer sizes 1MB, 100KB and 50 KB.

Figure 5(a) shows the affect of increasing μ on **average sensor buffer occupancy**.

As expected, with increasing μ the average buffer occupancy decreases. This is because when MULEs come more frequently there is less amount of data generated between two arrivals. Further, interestingly, SB does not have much effect on buffer occupancy except when the MULE arrival rate is small. This can be explained in the following manner. When μ is small, large amount of data is generated between two MULE arrivals. If SB is small, the data would be dropped and buffer occupancy will stay low. However, if SB is large, the data would be stored in the buffer and the buffer occupancy increases. Infact, for large μ, buffer occupancy is approximately $\frac{\lambda}{\mu}$ (Equation 9).

Figure 5(b) shows the effect of increasing μ on the **data success ratio (DSR)**.

With increasing μ, the DSR increases sharply eventually reaching one. This is because when μ is large, the buffer occupancy decreases and therefore less data is dropped. The arrow on each curve shows the minimum value of μ required for stability of the queuing system. This is $\frac{\lambda}{SB}$ (Equation 2). If $\mu = \frac{\lambda}{SB}$, the DSR is very low (around 0.6). Therefore, μ should be much larger than the minimum required. For our experiments, $\mu > 5 \times \frac{\lambda}{SB}$ resulted in DSR greater than .95.

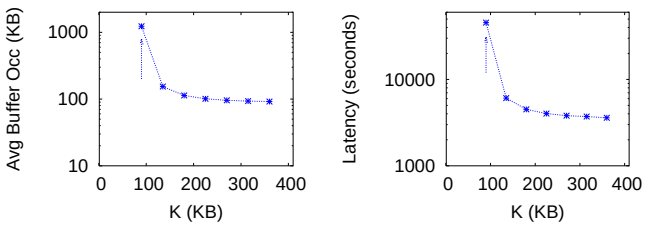
The DSR is higher, when SB is larger. This is expected because when SB is large, less data is dropped. Similar to scaling of μ, good DSR was achieved when SB was chosen such that $\mu > 5 \times \frac{\lambda}{SB}$. In general, one can increase DSR by either increasing μ or SB.

Figure 5(a) shows the effect of increasing μ on **latency**.

As mentioned before, only the queuing delay is considered. Since K is large, the queuing delay is simply the residual life of the MULE arrival process. Since the MULE arrival is poisson, the residual life is $\frac{1}{\mu}$ (by Corollary -B.1.1 in appendix -B). Therefore with increasing μ the latency decreases. Additionally, SB has no impact on latency, thus the three lines coincide.

A.2 Effect of K

Figure 6(a) and 6(b) shows the effect of increasing K on the average buffer occupancy and the latency respectively (note that the y-axis is logscale). We chose μ as 1 per hour and relatively large SB of 1MB. Since SB is large, the DSR is always close to one and is not shown. The arrow on the plots correspond to



(a) Average Buffer Occupancy (b) Latency

Fig. 6. Effect of scaling K. (a) shows average buffer occupancy and (b) shows latency

the minimum value of K for a stable queue. This is the ratio $\frac{\lambda}{\mu}$ (Equation 1) and for our plot it is 90 KB. We make the following observations:

1. When K is small, both buffer occupancy and latency is large. This is because a sensor cannot transfer all the data in the queue to a MULE during a single contact. This increases the average buffer occupancy. Latency is also increased because a data unit has to wait for multiple MULEs to arrive before it can be served.

2. As K is increased, there is a sharp decrease in both the buffer occupancy and the latency initially. For example, when K is doubled from its minimum value (90KB) the average buffer occupancy decreases by a factor of ten.

3. Increasing K beyond a certain limit does not effect performance. This follows by observing the flat region of the plots. Intuitively, this is because K only needs to be large enough so as to absorb the occasional burst in the sensor buffer. For our experiments, we found that $K = 3 \times \frac{\lambda}{\mu}$ was sufficient to be in the flat region.

A.3 Summary

Table VIII-A.3 summarizes the qualitative relationship between the different parameters and the metrics. [5] We also find that:

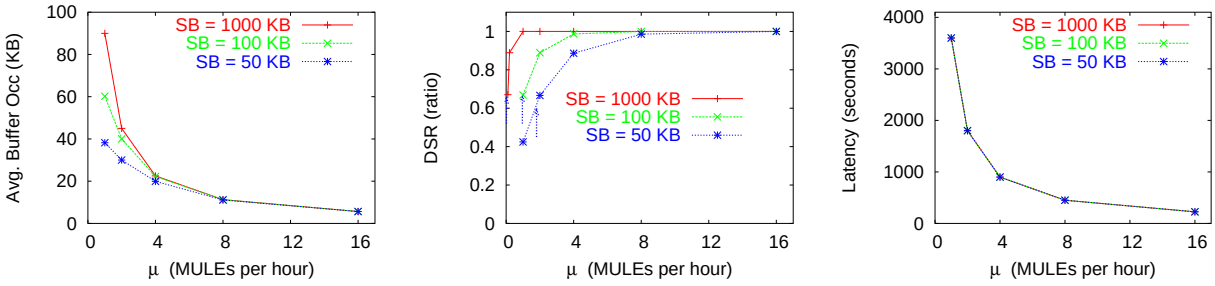
- DSR is less than 60% if the parameters are chosen such that the stability condition is just met.
- DSR can be made close to one by increasing μ or SB. When K is large, choosing SB and μ such that $\mu B > 5\lambda$ resulted in a DSR greater than 95%.
- When K equals $\frac{\lambda}{\mu}$, the minimum value, the sensor buffer occupancy is quite large (as compared to $\frac{\lambda}{\mu}$ as in Equation 9). However, the performance improves sharply by increasing K initially and eventually saturates when $K > 3 \times \frac{\lambda}{\mu}$.

B. Effect of Mobility Model

This section investigates how the choice of MULE's mobility model affects performance. This is done by considering MULEs arrival distribution resulting from the following mobility models:

1. **Poisson arrival of MULEs:** This is the default model discussed so far. MULEs arrival at a sensor is described by a Poisson process. This makes the analysis tractable and as mentioned

[5] Effect of increasing λ is the same as decreasing μ.



(a) Average Buffer Occupancy (b) DSR (c) Latency

Fig. 5. Effect of scaling μ on performance metrics. (a) shows average buffer occupancy. (b) shows DSR and (c) shows latency.

Parameters	Performance Metrics		
	Buffer Occ	DSR	Latency
$\mu \uparrow$	$\downarrow$	$\uparrow$	$\downarrow$
$SB \uparrow$	–	$\uparrow$	–
$K \uparrow$	$\downarrow$	$\uparrow$	$\downarrow$
$\lambda \uparrow$	$\uparrow$	$\downarrow$	$\uparrow$

TABLE I

EFFECT OF PARAMETERS ON PERFORMANCE. $\uparrow$INDICATES AN INCREASE IN THE QUANTITY. $\downarrow$INDICATES A DECREASE IN THE QUANTITY. – INDICATES NO EFFECT.

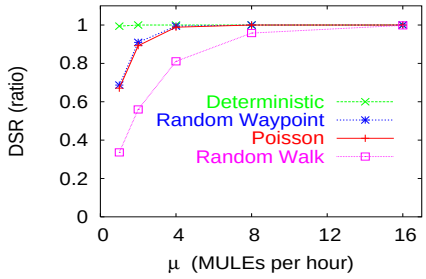


Fig. 7. Variation of DSR with μ for different mobility models

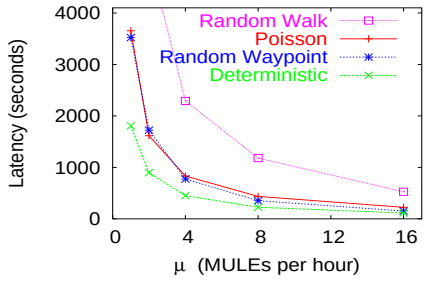


Fig. 8. Variation of latency with μ for different mobility models

in Section VII-C.1 by the virtue of the Palm-Khintchine theorem provides a good model for some applications.

2. **Deterministic arrival of MULEs:** MULEs arrive at a fixed interval at sensors. For example, when MULEs are scheduled city buses in a traffic-monitoring application or are controlled entities such as robots.

3. **Random motion:** Here MULEs are independent entities moving randomly over a two dimensional plane. This is suitable for applications such as wild life monitoring where MULEs are animals. We consider the following two models here [37]:

Random-Waypoint: MULEs randomly choose their destination and move towards it. On reaching the destination, a new destination is chosen randomly. Random waypoint is widely used to model the motion of random entities in mobile ad-hoc networks.

Manhattan or Random-Walk: MULEs randomly choose a direction and move in that direction. On reaching an intersection, a new direction is chosen randomly. Our topology is a two dimensional grid and on reaching an intersection either of the four directions (east, west, north or south) is chosen with equal probability. This is typically used for emulating the movement of mobile nodes on streets.

We do not have the MULEs arrival distribution (X^s) in closed form when the MULE motion is random. Therefore simulation is used to obtain the results. A custom simulator was written to simulate MULEs' motion and the queuing model presented in the modeling section VI. The details are omitted due to space constraints.

The underlying topology used in simulations had dimensions 2km*2km. The topology was divided into square grids each of size 25m*25m [6]. The MULE velocity was set to be at 10 m/s. For a fair comparison across multiple mobility models, the number of MULEs were chosen such that the MULE arrival rate (μ) was same across different models. The data generation rate was 90KB/s. K and SB were fixed at 100 KB and only μ was varied.

Figure 7 and 8 shows the DSR and the latency for different mobility models as μ scales. In all cases as μ increases, the DSR increases and the latency decreases. The performance is best when the MULE arrival is deterministic and worst under the manhattan model. The performance of random-waypoint model closely matches that of poisson model.

To understand this behavior, consider the coefficient of variation (CVR) for different mobility models as shown in Table II[7]. CVR gives an idea of the burstiness of MULEs arrival. Large CVR means that the MULE arrival pattern is more bursty and vice-versa. For example, for deterministic arrivals the CVR is zero because the MULEs arrive at fixed intervals.

[6]We tried few variations in the topology dimensions and qualitatively similar results were obtained

[7]Coeffi cient of variation is the ratio of the standard deviation to the mean. In general, for random motions it varied slightly with μ. Values shown are for μ = 8 MULEs/Hour

Mobility Model	Poisson	Determ-inistic	Waypoint	Manhattan
CVR	1.0	0.0	0.75	2.1

TABLE II
COEFFICIENT OF VARIATION (CVR) FOR MULE INTER-ARRIVAL DISTRIBUTION FOR DIFFERENT MOBILITY MODELS

Now, the performance would be better when the MULEs arrive at regular interval than in bursts (assuming same μ). This is because when the MULE arrival pattern is bursty, relatively longer periods exist when no MULE arrives. This can cause the sensor buffer overflow and reduce the DSR. This also affects latency because latency increases with the variance as discussed in the latency analysis (Results 3, 4).

C. Energy Savings

This section compares the energy consumption in the MULE architecture to an ad-hoc network. Only consider the energy required for sending and receiving data is considered. Energy consumed in idle radio listening is dictated by sensor's duty cycle and can be made comparable for both approaches. The following metrics are used:

Average Energy Ratio: This is the ratio of the average energy consumed per unit time at a sensor in the ad-hoc network to the average energy consumed in the MULE architecture.

Hotspot Ratio: This is the ratio of *hotspot usage* in the ad-hoc network to the *hotspot usage* in the MULE architecture. *Hotspot Usage* is defined as the maximum energy consumed by any sensor. As discussed in section III, hotspot usage gauges the network lifetime.

Before presenting results, we discuss how we compute energy requirements for the above two approaches. The following model is used for communication energy [28].

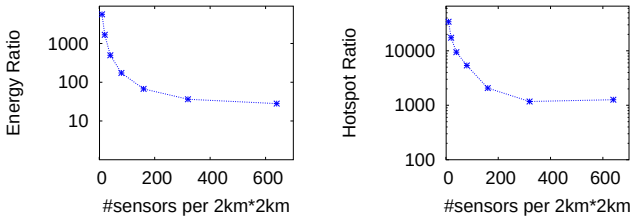
$$p_t = (\alpha_{11} + \alpha_2(d)^l)$$
$$p_r = (\alpha_{12})$$

p_t is the energy dissipated to transmit 1 bit of data to a node at a distance d. p_r is the energy dissipated to receive one bit of data. l is the path loss index and α's are positive constants. Here, α_{11} = 45 nJ, α_{12} = 135 nJ, α_2 = 10pJ/m^2 (l=2) or .0001 pJ/m^4($l = 4$), if $d < 87m$ $l = 2$, else $l = 4$.

Energy Requirements in the MULE architecture: In the MULE architecture, a sensor communicates data only to a MULE within range r. [8] Therefore transmit energy per bit (per sensor) is simply $\alpha_{11} + \alpha_2(r)^l$. We will take r as 25m.

Energy Requirements in an ad-hoc network: This depends on the sensor network topology and the routing protocol. A sensor communicates data to a nearby sensor towards an access-point and the forwarding continues until the data reaches the access-point. The exact choice of route depends on the routing algorithm used [7]. We route the data through the minimum energy path [7]. This optimizes the average energy consumption (within the ad-hoc network domain); though may not optimize the hotspot usage. Energy requirements for route maintenance are ignored, therefore, the energy computed here is only a lower bound on the overall energy requirements.

A custom simulator was used to compute the energy requirements using the above methodology. Sensor network topologies were generated by placing sensors randomly over a plane of dimensions 2km*2km. There was one access-point and was placed at a cornet of the topology. Results were averaged over 100 random simulations.



(a) Avg. Energy Ratio (b) Hotspot Ratio

Fig. 9. Energy comparison of MULE vs Ad-hoc network approach as a function of sensor density

Results

Figure 9(a) shows the *Average Energy Ratio* as a function of the sensor density.

When the sensor density is low, the MULE architecture has over a factor of 100 less average energy consumption than the ad-hoc network approach. This is primarily because with few sensors the average distance between two sensors is large. Since the communication energy increases as the fourth power of distance, this leads to enormous energy consumption in the ad-hoc network. The benefits decrease as the sensor density is increased and eventually saturate with the average energy ratio around ten. This highlights that even for high sensor density the MULE architecture is more efficient. This is because in the ad-hoc network the data traverses multiple hops. So, although the energy required per hop is a dense ad-hoc network is comparable to the MULE architecture, the total energy required is much more.

Figure 9(b) shows the *Hotspot Ratio* as a function of the sensor density. Same trend as in the Average Energy Ratio is observed. The Hotspot usage in the ad-hoc network is much more than the MULE architecture. This indicates that the life-time in the MULE architecture will be much longer than the ad-hoc network.

Additionally, the Hotspot Ratio is over an order of magnitude higher than the Average Energy Ratio. This indicates that in an ad-hoc network the maximum energy consumption (measure of the hotspot usage) is much more than the average energy consumption in it[9]. This is because the sensors near the access-points have to forward much more data than others. On the other hand in the MULE architecture all sensors have the same energy consumption.

These results are not surprising and are somewhat biased because in the MULE architecture there is an additional energy

[8] It is assumed that there are no buffer overflows and all the sensed data is transferred to MULEs.

[9] This was also verified explicitly by computing the distribution of the energy consumption at various sensors in the ad-hoc network. The maximum energy was found to be much more than the average energy.

consumption at the MULEs. However, MULEs are assumed to be entities with renewable energy whereas sensors are energy constrained and the primarily bottleneck of the system.

IX. Enhancements

The basic MULE system offers several interesting areas of investigation to increase its applicability and performance. In this section we outline and discuss some of these aspects.

- **Reducing sensor duty cycle**

Reducing sensor duty cycle saves energy, but as discussed in section VII-E, also affects the system performance as the sensors may not discover a nearby MULE. However, the probability of discovering a MULE can be improved by increasing the contact time (Equation 12), thereby allowing a reduction in the duty cycle without affecting performance.

The basic idea involves MULEs using longer range radios to transmit discovery messages. Sensors then have the opportunity to hear the message for a longer period of time, thereby increasing the effective contact time (section VII-D). Once the sensor hears the discovery message, it can keep the radio-on and wait for the MULE to come within the communication range of the sensor radio.

However, there is a possibility that a MULE will take a path that is outside the communication range of the sensor but is within the range needed for the long-range discovery radio. This causes the sensor to awake unnecessarily. This can be modeled by a probability of false alarm and can be traded off with the probability of not discovering a MULE. This strategy also increases the energy consumption of MULEs.

Application specific knowledge can also be used to reduce duty cycle. For example, if a sensor is aware of a MULE's arrival schedule then it can simply start listening at an appropriate time. Another possibility is to adapt the duty cycle based on buffer occupancy. Here a sensor keeps the radio totally off until a fraction of sensor's buffer get filled. After that the sensor switches to the normal mode of periodic listening. This would work particularly well when a sensor has relatively large buffers and latency is not important.

- **End to End Reliability** A simple method to achieve reliability is to incorporate acknowledgements (acks). This would cause sensors to wait for an ack before deleting the data from their buffers. This increases the sensor buffer requirements and places an emphasis on quick delivery of acks.

The main challenge is to determine when the sensors should retransmit their data. There is a trade-off, as retransmitting data too early may cause unnecessary transmissions that increase energy consumption; whereas, delaying retransmission may lead to buffer overflow and increased latency. The problem is particularly acute because of large and highly variable latencies. To improve the delivery of acks, a MULE can carry all acks, i.e, whenever a MULE reaches an access-point it picks up all acks, including acks for the data delivered earlier by other MULEs. This results in increased, but likely manageable, buffer occupancy at MULEs.

- **MULE to MULE communication** It is possible that a particular MULE may not be in the range of an access-point often enough. This can be addressed by having MULEs, that come within range of each other, exchange data. Later, if any of the MULEs reach an access-point the data will be delivered. The issues here are similar to epidemic routing [11, 3]. The primary trade-off is reduced latency at the expense of increased energy and memory consumption on the MULEs. This also poses the question of which data to exchange when MULEs meet.
- **Unreachable Sensors: Sensor Islands** This addresses situations in which some sensors may be out of reach of a MULE; for example in a forest. A sensor island is defined as a group of sensor(s) which are physically close enough to each other to form an ad-hoc network, such that at least one sensor comes within the range of a MULE occasionally. These reachable sensors act as transfer points so that the unreachable sensor's data may be transmitted to a MULE. The challenge here is to identify appropriate islands and the role of each sensor in that island without causing a hotspot detrimental to network lifetime.

X. Conclusion

This paper describes the MULE architecture, a three-tiered design, to enable energy efficient data collection in large and sparse wireless sensor networks. The key idea is to exploit the presence of mobile nodes in the environment by using them as forwarding agents. This approach extends the lifetime of the network by minimizing the communication responsibility of the resource-constrained sensors. A detailed analytical model based on queuing theory was presented that aids in understanding the limits and performance trade-offs inherent in the MULE architecture. Through simulation we confirmed that energy savings of up to two-orders of magnitude (and even larger increases in network lifetime) can be achieved with MULEs as compared to the traditional ad-hoc network approach. We also address the issue of efficient sensor discovery by MULEs.

Our MULE architecture is limited to applications that have some specific properties. First, the application must have an appropriate mobile agent that can be scaled easily to the requirements of data delivery. Second, it is applicable only for applications that do not have real-time requirements

This work is only a first step in understanding the feasibility of using mobility in sensor networks. It is clear that much more work remains to be done to fully understand the cost-effectiveness of this approach. We plan to investigate some of the enhancements discussed earlier, such as reliability and using MULE-to-MULE communication. Issues surrounding naming, network layer, and end-to-end connectivity semantics also needs to be addressed. Here we hope to leverage work from a recently proposed network architecture called the Delay Tolerant Network [38]. Other directions include a more detailed simulation and specific application experiments to demonstrate the feasibility of the MULE approach.

Appendices

A. Glossary of notation and symbols

SB	The total buffer capacity on a sensor (in bytes)
λ	Mean data generation rate
γ	Duty cycle of sensor. Fraction of time a sensor is listening to discover MULEs

DT	Discovery Time. Time to discover a sensor by a MULE
CT	Contact Time. The average amount of time the MULE is in the radio range of sensor
Q	Random variable denoting the queue length at a particular sensor when a MULE arrives
P_j	Probability that Q equals j
K	Amount of data (in bytes) that can be transferred between a MULE and a sensor during one contact
r	Radio range within which sensor and MULE can communicate
$\{U(t), t \geq 0\}$	The renewal process counting the total amount of data generated at a given sensor till time t. Also called the data generation process
$\{S(t), t \geq 0\}$	The renewal process counting the total number of MULEs that have visited a given sensor till time t
X^s	Random variable denoting the inter-arrival time for renewal process $\{S(t)\}$
μ	Average MULE arrival rate at a sensor
σ_{ms}	Variance of X^s
$E[B^{no}]$	Average number of MULEs that arrive at a sensor while a data unit is in the queue. $E[B^{no}]$ excludes the MULE which serves the data unit itself
$\{R(t), t \geq 0\}$	The renewal process counting the number of access points that a MULE has visited till time t
μ_r	Average rate at which a MULE visits access points
σ_r	Variance of the inter-arrival time distribution of $\{R(t), t \geq 0\}$
DSR	Data Success Ratio
W^q	Average queueing delay
W^m	Average time spent by data on MULE
W	Average latency

B. Residual life theorem

Consider a renewal process $\{C(t), t \geq 0\}$. Residual life $(r(t))$ of $C(t)$ is defined as the time measured from t to the next renewal instant after time t. The average residual life is $E[r(t)]$ as $t \to \infty$.

Theorem -B.1: Consider a renewal process $\{C(t), t \geq 0\}$. Let μ_c be its average renewal rate and σ_c be the variance of its inter-arrival time distribution. Then, the average residual life $E[r(t)]$ is (Chapter 7 of [36]),

$$E[r(t)] = \frac{\mu_c^2 \sigma_c + 1}{2\mu_c} \tag{14}$$

Corollary -B.1.1: As a special case, if the process $\{C(t), t \geq 0\}$ is a poisson process, the average residual life simplifies to,

$$E[r(t)] = \frac{1}{\mu_c} \tag{15}$$

This is because for a poisson process the distribution of inter-arrival times is exponential with mean $\frac{1}{\mu_c}$. And for an exponential distribution, $\sigma_c = \frac{1}{\mu_c^2}$.

C. Proof of result for DSR. Result 2

***Result* 2**: Data Success Ratio (DSR) is given by:

$$\begin{aligned} DSR &= \frac{\mu E[min(K, Q)]}{\lambda} \\ &= \frac{\mu(\sum_{j=0}^{K} jP_j + \sum_{j=K+1}^{SB} KP_j)}{\lambda} \end{aligned}$$

Proof: DSR is the ratio of data delivered to the access-points to the amount of data generated in time t as $t \to \infty$. By our assumptions once a MULE picks up the data it is delivered to the access-point. Therefore, DSR is the ratio of the data picked up by the MULEs in time t to the total data generated in time t.

$$DSR = \lim_{t\to\infty} \frac{P(t)}{U(t)} = \lim_{t\to\infty} \frac{P(t)}{t} (\lim_{t\to\infty} \frac{U(t)}{t})^{-1}$$

Here $U(t)$ is the total amount of data generated at the sensor and $P(t)$ is the total amount of data picked up by the MULEs. Also recall, $S(t)$ is the number of arrivals of MULEs in time t.

Now,

$$\lim_{t\to\infty} \frac{P(t)}{t} = \lim_{t\to\infty} \frac{P(t)}{S(t)} \lim_{t\to\infty} \frac{S(t)}{t}$$

By definition, $\lim_{t\to\infty} \frac{S(t)}{t} = \mu$ and $\lim_{t\to\infty} \frac{U(t)}{t} = \lambda$.

The term $\frac{P(t)}{S(t)}$ represents the average amount of data transferred when a MULE visits the sensor. Let L be the amount of data picked up by a MULE at the sensor. Then,

$$\lim_{t\to\infty} \frac{P(t)}{S(t)} = E[L]$$

Since only a maximum of K data units can be transferred,

$$L = min(K, Q)$$

Since, P_j is the probability Q equals j,

$$E[L] = \sum_{j=0}^{K} jP_j + \sum_{j=K+1}^{SB} KP_j$$

Putting everything together, we get the result. ■

D. Expression for $E[B^{no}]$

$$E[B^{no}] = \sum_{i=0}^{\lceil \frac{SB}{K} \rceil - 1} i \sum_{j=(iK)}^{iK+K-1} P_j^e$$

Proof: $E[B^{no}]$ is the average number of MULEs that arrive at a sensor while a data unit is in the queue. This depends on the distribution of queue length at the instant a new data is accepted in the queue. To compute this we define P_j^e which is the probability that the queue length is j at the instant a new packet

is accepted in the queue (excluding the new data unit itself). P_j^e can be related to P_j by (Theorem 4.1 of [34])

$$P_j^e = \sum_{i=j+1}^{min(j+K,SB)} \frac{P_i}{E(L)} \quad 0 \leq j < SB$$
$$P_j^e = 0 \qquad j \geq SB$$

The B^{no} of a new data unit is i iff the queue length (exluding the packet itself) is between iK and $iK + K - 1$. This is because a single MULE arrival removes K data units from the queue. This gives,

$$E[B^{no}] = \sum_{i=0}^{\lceil \frac{SB}{K} \rceil - 1} i \sum_{j=(iK)}^{iK+K-1} P_j^e$$

■

REFERENCES

[1] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David E. Culler, and Kristofer S. J. Pister, "System architecture directions for networked sensors," in *ASPLOS*, 2000, pp. 93–104.

[2] G. J. Pottie and W.J. Kaiser, "Wireless integrated network sensors," in *Communication of ACM*, May 2000.

[3] P. Juang, H. Oki, Y. Wang, M. Martonosi, and D. Rubenstein L. Peh, "Energy-effi cient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet," in *ASPLOS-X*, October 2002.

[4] Allan Beaufour, Martin Leopold, and Phillipe Bonnet, "Smart-tag based data dissemination," in *WSNA*, Atlanta, GA, Sept. 2002.

[5] Alan Mainwaring, Joseph Polastre, Robert Szewczyk, David Culler, and John Anderson, "Wireless sensor networks for habitat monitoring," in *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, Atlanta, GA, Sept. 2002.

[6] Deborah Estrin and Ramesh Govindan, "Next century challenges: Scalable coordination in sensor networkd," in *ACM/IEEE MobiCom*, 1999.

[7] Suresh Singh, Mike Woo, and C. S. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *Mobicom*. 1998, pp. 181–190, ACM Press.

[8] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad-hoc wireless networks," in *IEEE/ACM Trans. on networking*, vol. 10, no. 4, Aug. 2002.

[9] Ioannis Chatzigiannakis, Sotiris Nikoletseas, and Paul Spirakis, "An effi - cient communication strategy for ad-hoc mobile networks," in *Proceedings of the twentieth annual ACM symposium on Principles of distributed computing*. 2001, pp. 320–322, ACM Press.

[10] Zong Da Chen, H.T.Kung, and Dario Vah, "Ad hoc relay wireless networks over moving vehicles on highways," in *MobiHoC,*, 2001.

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

[12] Qun Li and Daniela Rus, "Sending messages to mobile users in disconnected ad-hoc wireless networks," in *Proceedings of the sixth annual international conference on Mobile computing and networking*. 2000, pp. 44–55, ACM Press.

[13] "Manatee," http://distlab.dk/manatee/.

[14] R.H Frenkiel, B. R Badrinath, Joan Borras, and Roy Yates, "Infostations challenge: Balancing cost and ubiquity in delivering wireless data," in *IEEE Personal Communications*, April 2000.

[15] S. Banerjee and A. Misra, "Minimum energy paths for reliable communication in multi-hop wireless networks," in *MobiHoC*, June 2002.

[16] Jae-Hwan Chang and Leandros Tassiulas, "Energy conserving routing in wireless ad-hoc networks," in *INFOCOM (1)*, 2000, pp. 22–31.

[17] Ya Xu, John Heidemann, and Deborah Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the seventh annual international conference on Mobile computing and networking*. 2001, pp. 70–84, ACM Press.

[18] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan, "Energy-effi cient communication protocol for wireless microsensor networks," in *HICSS*, 2000.

[19] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris, "Span: An energy-effi cient coordination algorithm for topology maintenance in ad hoc wireless networks," in *Mobicom*, Rome, Italy, July 2001, pp. 85–96.

[20] Akyildiz I. F., W. Sankarasubramaniam Su, and Y.and Cayirci, "A survey on sensor networks," in *IEEE Communications Magazine*, August 2002.

[21] C. Santivanez, B. McDonald, I. Stavrakakis, and R. Ramanathan, "On the Scalability of Ad Hoc Routing Protocols," in *IEEE INFOCOM 2002*, New York, NY, June 23–27 2002.

[22] M. Bhardwaj, A. Chandrakasan, and T. Garnett, "Bounding the lifetime of sensor networks via optimal role assignments," in *Infocom*, 2002.

[23] Qun Li, Javed Aslam, and Daniela Rus, "Online power-aware routing in wireless ad-hoc networks," in *Proceedings of the seventh annual international conference on Mobile computing and networking*. 2001, pp. 97–107, ACM Press.

[24] Curt Schurgers, Vlasios Tsiatsis, Saurabh Ganeriwal, and Mani Srivastava, "Topology management for sensor networks: Exploiting latency and density," in *MobiHoc*, 2002.

[25] Fan Ye, Haiyun Luo, Jerry Cheng, Songwu Lu, and Lixia Zhang, "A two-tier data dissemination model for large-scale wireless sensor networks," in *Proceedings of the eighth annual international conference on Mobile computing and networking*. 2002, pp. 148–159, ACM Press.

[26] Scott Shenker, Sylvia Ratnasamy, Brad Karp, Deborah Estrin, and Ramesh Govindan, "Data-centric storage in sensornets," in *First Workshop on Hot Topics in Networks (HotNets-I)*, Princeton, NJ, 2002.

[27] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-effi cient communication protocol for wireless microsensor networks," in *Hawaii Int'l. Conf. on system sciences*, 2000.

[28] W. Heinzelman, "Application-specifi c protocol architectures for wireless networks, ph.d. thesis, mit," 2000.

[29] Ivan Stojmenovic and Xu Lin, "Power aware localized routing in wireless networks," in *IEEE Transactions on Parallel and Distributed Syste*, VOL. 12, NO. 11, Nov 2001.

[30] Christian Bettstetter, "On the minimum node degree and connectivity of a wireless multihop network," in *MobiHoc*. 2002, pp. 80–91, ACM Press.

[31] Yu-Chee Tseng Chih-Shun, "Power-saving protocols for ieee 802.11-based multi-hop ad hoc networks," in *Infocom*, 2002.

[32] Rong Zheng and Robin Kravets, "On-demand power management for ad hoc networks," in *Infocom*, 2003.

[33] M.L. Chaudhry aand J.G.C. Templeton, *A First Course in Bulk Queues*, John Wiley and Sons, 1983.

[34] G. Hebuterne and C. Rosenberg, "Arrival and departures in the general bulk-service system," in *Naval Research Logistics, John Wiley, ed-Vol45*, 1998.

[35] D.P. Heyman and M.J. Sobel, *Stochastic Models in Operations Research (Vol. 1)*, McGraw Hill, NY, 1982.

[36] Sheldon M. Ross, *Introduction to Probability Models*, Academic Press, 2001.

[37] F. Bai, N. Sadagopan, and A. Helmy, "Important: A framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks," in *Infocom*, 2003.

[38] Kevin Fall, "A delay tolerant network architecture for challenged internet," Tech. Rep. IRB-TR-03-003, Intel Research Berkeley, 2003.