Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet

Philo Juang, Hidekazu Oki, Yong Wang[†],
Margaret Martonosi, Li-Shiuan Peh, and Daniel Rubenstein[‡]

Depts. of Electrical Engineering, [†]Computer Science, and [‡]Ecology and Evol. Biology
Princeton University

ABSTRACT

Over the past decade, mobile computing and wireless communication have become increasingly important drivers of many new computing applications. The field of wireless sensor networks particularly focuses on applications involving autonomous use of compute, sensing, and wireless communication devices for both scientific and commercial purposes. This paper examines the research decisions and design tradeoffs that arise when applying wireless peer-to-peer networking techniques in a mobile sensor network designed to support wildlife tracking for biology research.

The ZebraNet system includes custom tracking collars (nodes) carried by animals under study across a large, wild area; the collars operate as a peer-to-peer network to deliver logged data back to researchers. The collars include global positioning system (GPS), Flash memory, wireless transceivers, and a small CPU; essentially each node is a small, wireless computing device. Since there is no cellular service or broadcast communication covering the region where animals are studied, ad hoc, peer-to-peer routing is needed. Although numerous ad hoc protocols exist, additional challenges arise because the researchers themselves are mobile and thus there is no fixed base station towards which to aim data. Overall, our goal is to use the least energy, storage, and other resources necessary to maintain a reliable system with a very high 'data homing' success rate. We plan to deploy a 30-node ZebraNet system at the Mpala Research Centre in central Kenya. More broadly, we believe that the domain-centric protocols and energy tradeoffs presented here for ZebraNet will have general applicability in other wireless and sensor applications.

1. INTRODUCTION

Mobile computing and wireless communication are high-growth areas in the computer/communications arena. An increasing wealth of compute capability is available in handheld systems, and improved support for wireless communication helps interconnect these mobile platforms with each other, as well as with tethered desktop computers or servers. The main focus of mobile computing has been on systems such as PDAs and telephones intended for direct human use. Research attention is increasingly focused, however, on systems with more limited human intervention; wireless sensor networks are a key example. This paper examines the research decisions and implementation choices inherent in designing mobile compute/communication nodes for ZebraNet, a wireless sensor network aimed at wildlife tracking.

In general, *sensor networks* are systems in which numerous compute and sensing devices are distributed within an environment to be studied. Sensor networks have been proposed for a range of engineering, scientific and defense applications. While some sensor networks have static sensor positions, we focus here on issues related to dynamic sensor networks with mobile nodes and wireless communication between them. In fact, in our system, the sensor nodes are tracking collars carried by the animals under study; wireless ad hoc networking techniques allow them to swap and store data in a peer-to-peer manner and to percolate it towards a mobile base station that sporadically traverses the area to upload data.

An increasing focus of biology and biocomplexity research has been on gathering data and observations on a range of species, with a goal of understanding their interactions and influences on each other. For example, it is important to know how human development into wilderness areas affects indigenous species there. It is also important to understand the migration patterns of wild animals and how they may be affected by changes in weather patterns or plant life, by introduction of non-native species, and by other influences. Learning such details about animals requires both detailed long-term position logs as well as other biometric data such as heart rate, body temperature, and frequency of feeding.

Despite the importance of detailed data on animal movements and their relationship to weather, human development and other patterns, insufficient data currently exists. Furthermore, data collection technology is also quite limited. For the most part, current wildlife tracking studies rely on fairly simple technology. For example, many studies rely on collaring a sample subset of animals with simple VHF transmitters [10]. Researchers periodically drive through (or fly over) an area with a receiver antenna, and listen for pings from previously-collared animals. Once an animal is found, researchers can observe its behavior and log its observed position. The limits to such studies should, however, be fairly apparent. First, data collection is infrequent and may miss many "interesting events". Second, data collection is often limited to daylight hours, but animal behavior

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ASPLOS X, 10/02, San Jose, CA, USA.
Copyright 2002 ACM ISBN 1-58113-574-2-02/0010 ...$5.00

and movements in nighttime hours can be quite different. Third and finally, data collection is impossible or severely limited for reclusive species that avoid human contact.

Because of the limitations on simple VHF-aided visual observations, more sophisticated trackers are slowly becoming available. The most sophisticated trackers currently commercially-available use global positioning systems (GPS) to track position and use satellite uploads to transfer data to a base station [4, 19, 27]. These systems, however, also suffer from significant limitations. The most sophisticated tracker currently available only keeps a log of 3000 position samples and no biometric data [19]. Because satellite uploads are slow and power-hungry, they can only be done infrequently. This limits how often position samples can be gathered without overflowing the 3000-entry log storage. Furthermore, downloads of data from the satellite to the researchers are both slow and expensive (researchers are charged by the bit), constraining the amount of data collected. Finally, these systems operate on batteries without solar recharge, so when power is drained, the system is useless unless it is retrieved, recharged, and re-deployed.

Framing wildlife tracking as a sensor networks problem, the ZebraNet project is building tracking nodes that include a low-power miniature GPS system with a user-programmable CPU, non-volatile storage for data logs, and radio transceivers for communicating either with other nodes or with a base station. One of the key tenets of ZebraNet is that the system should work in arbitrary wilderness locations; we do not assume the presence of fixed antenna towers or cellular telephone service. The system therefore uses peer-to-peer data swaps to move the data around; periodic researcher drive-bys (or fly-overs) can then collect logged data from many animals despite encountering relatively few within range. While ad hoc sensor networks have been widely studied in the abstract, much less has been published about the characteristics of mobile sensor networks with mobile base stations and relatively few studies focus on building real systems. In particular, this paper offers several unique contributions:

- First, we believe we are the first to study protocols for mobile sensor networks in which the "base" station is also mobile. In our case, we presume that researchers will upload data while driving or flying by the region. And in fact, the base station is available only sporadically, when researchers are out driving a data-collection loop.

- Second, zebra-tracking is a domain in which the node mobility models are largely unknown, and in fact are ultimately the research goal. Understanding how, why, and when zebras undertake long-term migrations is the most pressing biological question for this work. In essence, we "bootstrap" mobility models by using current, less well-refined biology data to design our early protocols, which can then be refined and adapted as the initial deployed system helps us learn about zebra movements, especially long-term migrations, in more detail.

- Like other sensor networks, ZebraNet's data collection has stylized communication patterns in which data can be cooperatively funneled towards a base station. We optimize our protocols for this "data-gathering" communication pattern and for the high degree of latency tolerance in this application domain.

- Finally, we examine energy tradeoffs in detail, using real system energy measurements for ZebraNet prototype hardware in operation.

In considering ZebraNet, a number of interesting research questions arise. How to make the communications protocol both effective and power-efficient? To what extent can we rely on ad hoc, peer-to-peer transfers in a sparsely-connected spatially-huge sensor network? And finally, how can we provide comprehensive tracking of a collection of animals, even if some of the animals are reclusive and rarely are close enough to humans to have their data logs uploaded directly? This paper gives quantitative explorations of the design decisions behind some of these questions. In addition, we give initial systems experiences and power measurements for our ZebraNet prototype. More broadly, by summarizing early experiences with ZebraNet, we feel that this paper offers protocol ideas that should be relevant to a wide selection of researchers in the wireless and ad hoc networking domain.

The remainder of this paper is structured as follows. Section 2 describes the problem domain and metrics of interest in more detail. While ultimately biologists wish to place ZebraNet-style nodes on a range of species in an ecosystem, our first goal is to develop a collar design and protocol that works well with zebras. For this reason, Section 3 discusses the social structures and movement patterns for zebras that we use when designing protocols and reasoning about well-suited mobility models for our application. Following this, Section 4 gives an overview of the ZebraNet tracking node and collar design, Section 5 discusses ZebraNet protocols, and Section 6 reports their effectiveness and energy efficiency. Section 7 relates our work to other projects in sensor networks, energy-efficient mobile system design and other domains. Finally, Section 8 summarizes our results, discusses our future plans, and offers conclusions.

2. ZEBRANET DESIGN GOALS

The ZebraNet project is a direct and ongoing collaboration between researchers in experimental computer systems and in wildlife biology. The wildlife biologists have articulated the tracker's overall design goals as:

- GPS position samples taken every three minutes.

- Detailed activity logs taken for 3 minutes every hour

- 1 year of operation without direct human intervention. (That is, we should not count on tranquilizing and re-collaring an animal more than once per year.)

- Operation over a wide range (hundreds or thousands of square kilometers) of open lands. We plan to deploy our system at the Mpala Research Centre in central Kenya [25].

- No fixed base stations, antennas, or cellular service. (Any unguarded equipment, large or small, is too likely to attract attention and unfortunately, vandalism.)

- While latency is not critical, a high success rate for *eventually* delivering all logged data is important.

- For a zebra collar, a weight limit of 3-5 lbs is recommended. Smaller animals may need even lower weight limits.

The three-minute duration between position samples is motivated by biological research that shows that the interval is long enough to record statistically-independent behavior and yet frequent enough to log sufficient data points over time [1]. In addition, once per hour, the unit will log detailed information for a duration of 3 full minutes. Ultimately, this detailed information might include several position estimates, temperature information, weather data, environmental data, and body movements that will serve as signatures of behavior; in our initial system here, however, we focus solely on position data.

Overall, the key goal is to deliver back to the researchers a very high fraction of the data collected over the months or years that the system is in operation. As a result, ZebraNet must be quite power-efficient, must be designed with adequate data log storage, and must be rugged to ensure reliability under tough conditions.

2.1 ZebraNet Problem Statement

Having stated above the biologists' design goals, we next turn to the implications of those goals on the engineer's task at hand.

The primary figure of merit for our designs is that the success rate at delivering position data to the researcher—a metric which we refer to as the *data homing* rate— should approach 100%. The engineering research problems arise from several issues.

For example, as shown in Section 4, weight limits on each node translate almost directly to computational energy limits. This is because the weight of the battery and solar panel dominates the total weight of a ZebraNet node. As a result, our collar and protocol design decisions must manage the number and size of data transmissions required. We must also make system design choices that limit the range of transmissions, since the required transmitter energy increases dramatically with the distance transmitted. Finally, we must limit the amount of storage needed to hold position logs. At roughly 6KB per day, a single animal's position data uses relatively little storage. But if many redundant copies are stored and swapped, the storage requirements can scale as $O(N^2)$. Although the energy cost of storage is small compared to that of transmissions, it still behooves us to develop a storage-efficient design.

Because of limited transceiver coverage and a base station only sporadically-available, ZebraNet must forward data through other nodes in a peer-to-peer manner and store redundant copies of position logs in other tracking nodes. Section 5 discusses our protocol experiments for operating in a system with mobile sensors and base station, as well as bandwidth and storage constraints.

Some of the key challenges in ZebraNet come from the spatial and temporal scale of the system. In terms of temporal scale, keeping a system running autonomously for months at a time is challenging; it requires significant design-time attention to both hardware and software reliability. We also plan work (not discussed here) to implement on-the-fly software updates which will facilitate bug fixes and parameter tuning after the collars are deployed. In terms of spatial scale, ZebraNet is also aggressive; it is the specific intent of our system to operäte over an area of hundreds or thousands of square kilometers. Because of the large distances involved and sparse sensor coverage, energy/connectivity tradeoffs become key.

The challenges and issues outlined here come together in a system design that tackles several open problems. Namely, ZebraNet's protocol promises good communication behavior on mobile sensors percolating data towards a mobile base station. Second, ZebraNet explores design issues for sensors that are more coarse-grained than many prior sensor proposals. The larger weight limits and storage budgets allow us to consider different protocols with improved leverage for sparsely-connected, physically-widespread sensors.

3. A DAY IN THE LIFE OF A ZEBRA

Mobility models are at the core of design decisions for many mobile networks. Mobility models help to abstract how fast and how often users (and therefore, wireless nodes) move, in what direction, and with what forces of attraction or repulsion. Likewise, to design ZebraNet, we also need to understand how the nodes will move, as this critically affects hardware, protocol and overall system design. Ultimately, we wish to deploy sets of ZebraNet collars on a range of species that share the same ecosystem: zebras, lions, wild dogs, and even large mammals such as elephants. This allows biologists to gather fundamental inter-species data that is currently woefully lacking. For this paper, however, we focus on zebras. We include this section to give specifics about zebra motion and social structure that impact our system design choices.

3.1 Social Structure and Collaring

Approximately 35,000 zebras range widely over the 40,000 square kilometers that comprise the Laikipia ecosystem of central Kenya. Understanding how they use the landscape requires collaring representative individuals and characterizing their fine-grained movements and behaviors over large scales. Fortunately, the social structure of some zebra species enables us to collar only males and yet still gather information on the ranging behavior of large subsets of the population.

Two species of zebras inhabit the Laikipia ecosystem. One, the Grevy's zebra (Equus grevyi) forms large loosely-bonded herds. The other, more common, Plains zebra (E. burchelli) forms tight-knit uni-male, multi-female breeding groups. These so-called "harems" are characterized by 4-5 females and their young offspring living in close association with a stallion for long periods of time, often many years. Females typically initiate movements but the male often adjusts the direction and speed of movement of the group [34]. Thus by collaring only the male we can effectively track the movement of 10-12 individuals, vastly reducing the number of collars required as we try to characterize the movements of entire plains zebra populations.

Although plains zebras live in tight-knit breeding groups, these groups often coalesce and form moderately stable long-term herds. Typically harem groups coalesce into herds at watering points before embarking on movements to new grazing grounds. En route, harems sometimes join or leave these herds depending on the structure of the habitat, the quality of the vegetation and the composition of individual harem groups [2]. Clearly, herds are more amorphous than the smaller harem groups, but they last longer than a mere temporary aggregation. Such dynamics present a challenging problem to ecologists trying to unravel their causes, but will actually assist ZebraNet in propagating position logs across the landscape towards a mobile base station.

3.2 Movement Patterns

Zebra movement can be characterized in terms of three main states: grazing, graze-walking, and fast-moving. Zebras spend most of their time grazing, both

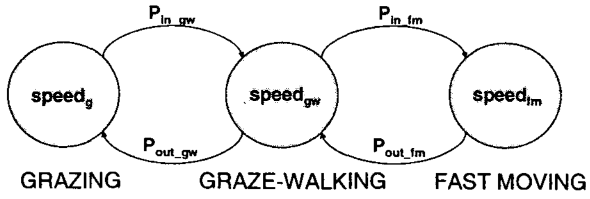


Figure 1: Three-tiered mobility model.

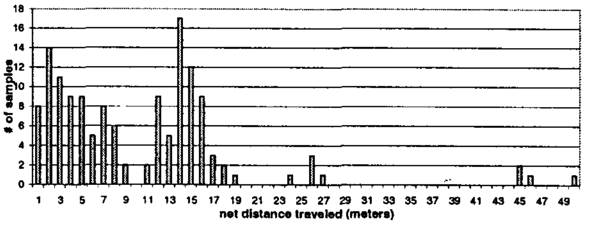


Figure 2: Distribution of zebra movements observed by field biologists.

day and night. Zebras prefer to graze in areas of short but rapidly-growing grasses. These areas offer high energetic gains and low risks of predation. While grazing on short grass swards, zebras typically exhibit low movement rates and high turning angles.

At other times, zebras walk deliberately, with heads lowered, clipping vegetation as they move. These latter movements are referred to as "graze walking" and are characterized by higher step rates and smaller turning angles than those for focused bouts of grazing.

Finally, either due to predators or because an area's vegetation has been exhausted, zebras will occasionally move much more quickly, for longer distances, with their heads raised because they are not grazing. We categorize this as the fast-moving state.

Figure 1 illustrates these three modes of zebra movement abstractly, with transition probabilities between them. The speed distributions in each mode and the probabilities of transitioning between each state are derived through feedback from biologists as described below.

Distance Moved. Figure 2 shows zebra movement data collected by field biologists [2]. The histogram shows how often different net movements were observed. Each data sample is net distance moved in a three-minute interval since the last observation. (The three-minute interval is chosen based on empirical biological studies that show its suitability for statistically-valid sampling of animal movements [1].) We define net distance moved as the net distance from the beginning of the three-minute interval to the end. That is, if a zebra moved ten meters from its original position and then came back again, all in three minutes, its net distance moved would be zero.

Because the data was collected by a stationary observer, Figure 2's data includes mostly grazing and graze walking observations. The two types of motion can be discerned by the bimodal nature of the distribution. The first mode, grazing, has a histogram peak graphed at 2m and a mean net-distance of 3.1m. The second mode, graze-walking, ranges from 10-20m, has a peak graphed at approximately 14m, and has a mean value of 13.0m. The few outliers in the distribution indicate points where the zebra may have sensed danger and fled.

Overall, it is clear that zebras tend to move very slowly; as they spend most of their time simply grazing, their net distance moved tends to be very small. This hints to us that routing protocols which intelligently exploit past link history information may be fruitful.

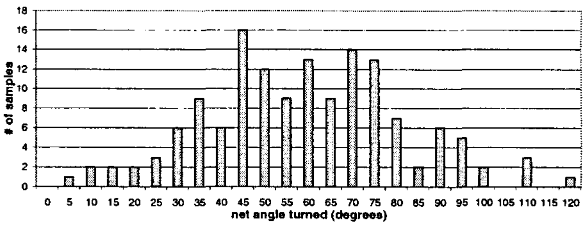


Figure 3: Distribution of zebra turning angles observed by field biologists.

Turning Angle. Another facet of movement is direction. Figure 3 gives field data on net turning angle. Similar to net distance moved, the net turning angle is defined as the absolute value of the angle between the start of the time interval and the end of the time interval. If the zebra moved 360 degrees within three minutes, its net turning angle would be zero. The maximum turning angle is therefore 180 degrees. We use these distance and direction histograms to guide mobility models for simulations in Section 6.

Water Sources and Drinking. Zebras are termed a water-dependent herbivore because they seek out water to drink on a daily basis. Again based on observations, our mobility models assume that zebras head for water sources about once per day. Once there, they drink relatively quickly. And once their thirst is satiated, their movement is again independent of the water source until the next day. We assume in our models that the sources of water are randomly distributed, and that thirsty zebras can easily (but not instantly) find their way to an adequate source.

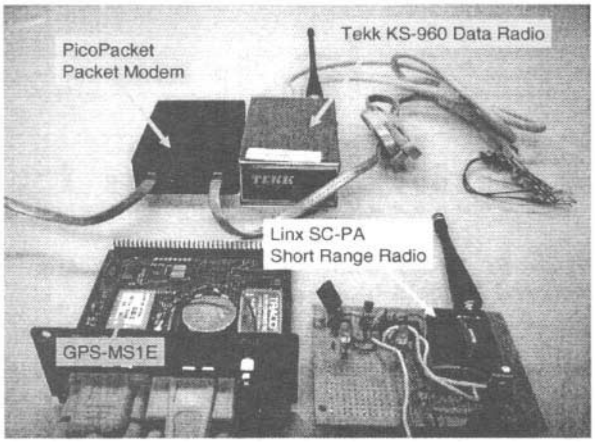
Sleep. Zebras tend not to have long periods of motionless sleep. Unlike carnivores, which are equipped with significant defense mechanisms, zebras rely on keeping watch and fleeing from predators. Therefore our models assume that zebras maintain their mobility pattern 24 hours a day.

4. COLLAR DESIGN

Figure 4 shows a photograph of the core of a ZebraNet prototype node: the evaluation board for the GPS-MS1E (containing a GPS, Flash RAM, and CPU), a short range radio, and a long range radio with its packet modem. (The photo does not show the packaging, batteries, solar array, and power management circuits.) This section gives an overview of the tracking collar node design. The block diagram in Figure 5 illustrates the different components and their interactions with one another.

To minimize the part count and overall size and weight of the system, we use a single-chip miniature GPS solution from μBlox: GPS-MS1E [5]. The GPS-MS1E is a 12-channel GPS receiver capable of getting a position update every second (though we get them less frequently). It has an integrated 20Mhz Hitachi SH1 32-bit microprocessor as well as I/O support. We use the SH1 for data capture and protocol control; it is the only programmable CPU in the ZebraNet node. The GPS-MS1E also has a built in 1MB Flash RAM module; 640KB is available for user data while the rest is used to store the firmware.

Using the GPS-MS1E's microprocessor, we periodically obtain the position coordinates and store them in



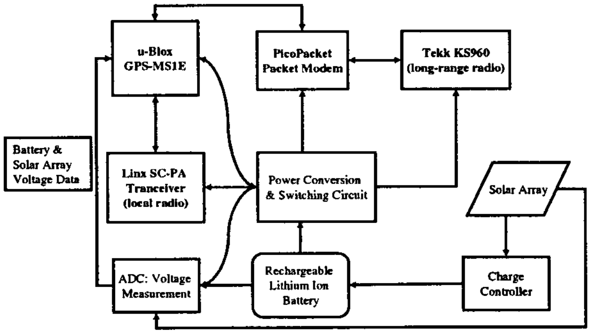


Figure 5: Block diagram of ZebraNet node design.

its on-board flash RAM. GPS readings are accurate to within 5-10m; this is more than sufficient for our purposes. Assuming that we store 30 coordinates per hour, each hour requires a little over 240 bytes of space. This implies that 640KB of storage is equivalent to approximately 110 collar-days worth of data. Furthermore, we plan to compress the data by representing most of the coordinates as offsets from two reference points per hour. Assuming a compression rate of about 36%, 640KB of Flash is then capable of storing 300 collar-days of data.

The processor also coordinates the communications over the two radios. We chose to use two radios so we can have broad control over tradeoffs in energy vs. communication range. First, the Linx Technologies SC-PA series [18] is a data radio with a range of only 100 meters but very low power consumption. Second, we use a slow but higher-power data radio and packet modem for longer-range (8km) transfers. The short-range radio is power-efficient for peer transfers when zebras are congregating by water sources, while the longer-range radio is necessary for communicating to the base station over the large area studied with relatively few tracking collars.

Short Range Radio Protocol. While the PicoPacket packet modem handles error correction, collision detection, and packetization for data sent on the long range Tekk data radio, the same for data transmission over the Linx radio must be performed by the ZebraNet firmware. Short range radio packets have a maximum size of 300 bytes and a 16 bit CRC provides error checking. The Linx radio also requires a MAC protocol since none is provided in the hardware. While many standard protocols such as Aloha, Slotted Aloha, CSMA, and MACA [16] are available, ZebraNet has requirements and resources that differ from typical wireless ad-hoc networks. When doing peer search, collars must avoid collisions by selecting designated senders one-at-a-time. Fortunately, we can implement a unique collision-avoidance protocol that takes advantage of the fact that GPS gives our networked system an extremely accurate and precise synchronized clock. (The system has networked timing with 30-50ns precision and 30ns RMS accuracy.) By broadcasting peer to peer search queries in non-overlapping predetermined time slots that repeat every 10 seconds or so, we can eliminate collisions. The minimum length of the time slots is dictated by the availability of CPU time and the time needed to switch between receive and transmit modes on the Linx radios. CPU availability is an issue since the GPS-MS1E CPU is also running time-critical tasks related to GPS tracking. Because of this, we work with 100-200ms time slots in our initial devices. This gives us collision-free operation for 50 or 100 collars, with 200ms and 100ms time slots respectively.

Wireless Networking Alternatives. Frequency range regulations affect the choice of radio for our prototyping purposes. For example, there are high-performance radios manufactured by MaxStream that operate in the 900Mhz and 2.4Ghz ISM bands that are license-free in the US [22]. In Kenya, however, we would have to use the shorter-range (and only very recently available) 2.4Ghz unit, the 24xstream. Nevertheless, it has a range of up to 4km line of sight with low power consumption (1.2W transmitting, 0.3W receiving). Because of its power advantages, we may switch in the near future to using this radio for long-range transmissions.

Finally, we also considered using an OEM wireless Ethernet (802.11b) module [35] instead of the short-range radios. The potential advantages of 802.11b would be very high data throughput and the ability to abstract away details of wireless communication including collision detection and avoidance, error detection, etc. There are disadvantages, however. The GPS-MS1E's serial ports only support speeds up to 33.8Kbps. This becomes such a severe bottleneck that unless we choose another I/O method, we lose most of the speed gain of Ethernet. Without the speed gain, our power requirements would go up by 16-25X per unit data transferred [35]. We could solve the I/O bottleneck by adding a separate microcontroller and storage, instead of relying on those provided by μBlox, but this would further increase the energy requirements of an Ethernet-based choice and also would increase per-node hardware costs, size, and complexity.

Energy Issues and Power Supply. Table 1 gives current consumption of the ZebraNet node operating in different modes. All figures in the table are current drains on the 3.6V power supply. The current figures are based on actual lab measurements of the current consumption of individual devices, but the aggregate current drains on the 3.6V supply for each mode were calculated assuming the use of 70% efficient DC-DC voltage converters with the appropriate output voltages powering devices that run on voltages higher than 3.6V. (The collars require DC-DC voltage converters with regulated outputs, especially for the long-range radio whose amperages are highly variable.)

Current drains range from a low of less than 1mA when the system is in stand-by mode (most of the time), to a high of 1.622A when the system is transmitting

Collar State	Device and Mode	Current drain of 3.6V supply
Stand-by	All	< 1mA
Position Sampling and Storage	GPS-MS1E, Active Antenna	177mA
Peer Discovery/Transfer Only	GPS-MS1E + Short-range	177mA
Base Discovery Only	GPS-MS1E + Long-range,	432mA
Simultaneous Peer and Base Search	GPS-MS1E + Short-range + Long-range	469mA
Transmitting Data to Base	GPS-MS1E +Long-range	**1622mA**

Table 1: Energy measurements for a ZebraNet node in different states of operation.

Item	Weight
μBlox GPS-MS1E Single-chip GPS/CPU	8 grams
Linx SC-PA Short-range Radio	20 grams
Long-range Radio and Packet Modem	296 grams
14 Sony Lithium-Ion Polymer Cells: (UP503759AH) 3.7v, 1AH cells	287 grams total
Solar Array - Unisolar USF5 flexible 5 watt	540 grams
Total	**1,151 grams (2.54 lbs)**

Table 2: Weight measurements for different components of a ZebraNet node.

using its packet modem and long-range radio. Our goal is to have a power supply system in which the battery is recharged from a solar array, but in which the battery can operate the system for 5 full days between recharges if needed. We conservatively assume that in those 5 days we will do the following:

- 30 position samples per hour, 24 hours every day.
- 6 (total) hours per day of searching for peer nodes and transferring data between them over low-power short-range radio.
- 3 hours of searching for the mobile base station using the long-range radio per day. To save energy, the 3 hours of base station search overlap in time with the 6 hours of peer search and peer transfer because in both modes, the relatively power hungry CPU must be on anyway.
- 640 kilobytes transmitted to mobile base station during 5 day period

To operate with the above assumptions, we need a 13.5 Ampere-hour battery with a voltage greater than or equal to 3.6 volts. A readily-available, easy-to-use lead-acid battery with appropriate capacity would weigh four pounds. Since this is too heavy, we are opting instead for Lithium-ion polymer cells, which have the highest energy density even among lithium ion cells. As indicated in Table 2, the required energy capacity with this battery technology will weigh about 287 grams or about 0.63lbs. Table 2 summarizes the weights for all the key components in a ZebraNet node. At this point, the heaviest single component is the flexible amorphous silicon solar cell array [37]. At 540 grams (1.18lbs), it contributes about half of the total collar weight. (Rigid solar cell arrays would be cheaper, lighter and have greater power generation efficiency, but flexible amorphous silicon arrays are better at withstanding rugged environments.)

Current Status. We have built two prototype copies of ZebraNet nodes, which are currently operational in the lab. In particular, they can now automatically sample GPS coordinates and store them in Flash RAM. In addition, they can use the short-range wireless radio to search for peers, and to exchange data with another collar.

5. PROTOCOL DESIGN

The goal in ZebraNet is to gather data collected at each collar back to the base station. Since not every collar is within range of the base station, data cannot be sent directly. Instead, it has to hop its way towards the base station, using other collars as intermediate hops. In ZebraNet, all nodes except the base station are data sources, while the base station alone is a data sink. This "data gathering" trait contrasts with the general end-to-end communication prevalent in many wired and wireless networks, where every node can be a source and/or sink.

In addition, ZebraNet nodes are mobile. The nodes move around almost constantly (albeit slowly). The base station is also mobile, depending on the route taken by researchers in their vehicles. Furthermore, the base station is only active some of the time, when researchers are driving around gathering data. In the duration that a base station is inactive, the network essentially has no known destination where data should be sent. These characteristics, coupled with the high latency tolerance of ZebraNet, call for specialized protocols.

5.1 Flooding Protocol

A simple approach to move data back to the base station is to flood data to all neighbors whenever they are discovered. Figure 6 shows the pseudo-code for the flooding protocol. If the nodes move extensively and meet a fair number of other nodes, then given enough time, data will eventually migrate back to the base. In this way, a high percentage of the data eventually makes it back to base.

The base station does not necessarily have to come into contact with all the nodes in the system; instead, coming into contact with just a few nodes may be enough. Indeed, it can be inferred that by identifying a few highly-interactive nodes, *i.e.* nodes that meet a large number of other nodes, we can collect a substantial amount of data readily.

While flooding can potentially return the highest success rate in a peer-to-peer network, the large amount of data flooded through the network can lead in some situations to exorbitant demands for network bandwidth, storage capacity, and energy.

```
At each scan for neighbors,
  if node is within range of the base station,
    send data to base station;
    delete this data, since it has successfully
      reached the base station;
  else
    send data to all neighbors;
```

Figure 6: Pseudo-code for the flooding protocol.

```
At each scan for neighbors,
  if node is within range of the base station,
    send data to base station;
    delete this data, since it has successfully
      reached the base station;
    increment hierarchy level;
  else
    check hierarchy levels of neighbors;
    send data to neighbor with highest level,
      breaking ties randomly;
    decay hierarchy level after D scans;
```

Figure 7: Pseudo-code for the history-based protocol.

5.2 History-based protocol

Rather than flooding data to all neighbors, we also consider a simple protocol that intelligently selects nodes to send to based on prior communication patterns. Naturally, a good target node is one that will ultimately relay the data to the base station. Our history-based protocol encodes the likelihood of a node being in range with the base station by assigning each node a hierarchy level based on its past success at transferring data to the base station. The higher the level, the higher the probability that this node is within range of the base station. The intuition behind this is that nodes that were previously within range of the base station will still be close by, so they will be able to relay the data back to the base station either directly (if they are still in range) or indirectly through minimal other nodes. This protocol thus biases the selection of a node based on history.

Each node remembers its own current hierarchy level. Each time a node scans for neighbors, it requests the hierarchy level of all its neighbors. It then sends the data it has collected to the neighbor with the highest hierarchy level, with ties randomly broken. When a node comes within range of the base station, its hierarchy level gets increased. Conversely, when a node is out-of-range from the base station, its hierarchy level gets decayed over time at a rate of one level per every D scans. That is, if D is 5, we decrement the hierarchy level by 1 every 5 consecutive scans where it is beyond the base station's range. At the start, all nodes start off at the same lowest hierarchy level of zero. The pseudocode of the proposed protocol is shown in Figure 7.

Clearly, the success of the history-based unicast routing protocol depends on the mobility of the base station and nodes. If the network changes very dynamically, a node that was previously near the base station may no longer be the best communication target. Then, the proposed protocol may mis-direct traffic frequently and get a poor homing success rate.

6. EXPERIMENTAL RESULTS

In this section we will describe our simulation environment and our protocol evaluation. We first present data on the ideal case; from here, we will constrain two major factors—storage and bandwidth—and then present energy tradeoffs between different protocols. Finally, we simulate our proposed design and show the results—success rate and energy consumption—of our design.

6.1 ZNetSim

Armed with facts and field observations about zebra behavior and reasonable assumptions of the terrain and operating characteristics of the Mpala Research Center in Kenya, we constructed a zebra mobility model and simulation environment for ZebraNet. Our simulator, ZNetSim, takes user-defined storage and bandwidth constraints, and returns two metrics: (i) success rate, which is the percentage of data that gets back to base, and (ii) energy consumption. We developed ZNetSim in C, and it currently stands at 5941 lines of code.

Mobility Models. At the start of each simulation, we randomly place 50 zebras and 10 water sources across a 20kmX20km map. As this is savanna, there are no major mountains or canyons that might hinder herd movements, animal interactions, or networking interactions, so we assume unobstructed communications. Once the zebras and water sources are placed, the map is set into motion. The zebra movements are based on the three-tier mobility model shown in Figure 2. Each zebra independently selects speed and turning angles such that aggregate three-minute movements match the distributions in Figures 2 and 3. Unless otherwise stated, the zebras move at a base speed of 0.017m/s when grazing, four times faster at 0.0723m/s when graze-walking, and nine times faster at 0.155m/s when fast-moving. Communication events are simulated on 30 second granularity.

Once per day at a random time, the simulated zebras become "thirsty." When thirsty, a zebra moves as if in "graze walking" mode—i.e. faster and more deliberately—towards the nearest watering hole. (We presume that they know the location of the nearest watering hole from any point on the simulated grid.) Finally, since field data indicates that zebra movements tend to be similar 24 hours per day, our simulator treats nighttime the same as daytime—an endless cycle of eating and walking. While predators do range across the areas under study, the zebra mortality rates due to predators are low enough that we ignore them for these simulations.

We compared ZNetSim's mobility model with that observed by biologists and found our distribution to match almost exactly with Figures 2 and 3, with the discrepancies being simply rounding error. The base station itself follows a rectangular route from (5km, 5km) to (15km, 15km) in the 20km by 20km map. The base moves three hours per day, between 2 p.m. and 5 p.m., and moves at 8m/s, or roughly 30km/h. Once three hours are up, it goes off-line immediately, but restarts the next day from this same location.

Simulation Methodology. Our simulations consist of four communication phases that occur within 30 minutes every two hours, *i.e.* from 12:00-12:30, 2:00-2:30, 4:00-4:30, etc, over an entire month. This timeline is arrived at due to the power constraints discussed in Section 4, as collars are limited to six hours per day of searching for peers and transferring data. The four phases are:

- Peer Discovery: All nodes first enter a mode where they use their short-range receivers to search for neighbors within range.
- Base Discovery: Likewise, nodes with a separate long-range radio will query to see if they are within

range of the base station. Since the nodes do not know when the base station will be available, base discovery is done from noon till midnight every day. This is typically overlapped with Peer Discovery to save power.

- Peer Transfer: Upon finding one or more nodes within range, one collar initiates data transfers. Once this node has finished its transfer, another node begins, till the end of the 30 minutes. The ordering of these collar selections is random in our simulator, but future protocols may try to optimize this order.
- Base Transfer: After successfully finding the base within range, collars upload all stored data to the base station. With our long range radio, we assume that total bandwidth can be shared, so all radios within range of the base can transfer at the same time, dividing the bandwidth equally. Once the data entries are transferred, they are deleted from the collar to free up storage.

We assume peer and base discoveries take 30 seconds and peer and base transfers are dependent on the available bandwidth and the amount of data to be transferred. In all transfers, nodes send their own data first before forwarding other nodes' data. Once the 30 minutes communication interval is up, all discoveries and transfers immediately cease. Unless otherwise mentioned, we use a single radio in all simulations. This lets us more clearly illustrate the effect of radio range on network performance. Finally, we note that for simplicity we ignore the irregular and asymmetric characteristics of radio ranges as discussed in [9].

Deletions with limited storage. With limited storage, a node prioritizes its own data over that collected from others. So, if a data point comes in and there is no free space to store it in memory, the node first deletes the oldest data point belonging to another node. If none are available, it will then delete its own oldest data point. In this way, the system prioritizes the most recent timestamped points; the data points that have been around the system the longest—and thus had the highest probability of being already transferred to base—are the first to be evicted to make room for newer incoming points. Similarly, a node's own data is always last to be evicted, and in that case only for newer points of itself.

Once a data point has been transmitted to base, it is added into a "delete list." The delete list is a data structure that indicates a particular point is now obsolete and can be erased. Like regular data points, delete lists are also transferred between nodes. Unlike regular data points, delete lists do not contain full data points. In peer to peer transfer, upon receiving a data point, if it is already in the delete list, it is discarded. In addition, once every hour, the nodes "scrub" their memories of data points in the delete list.

6.2 Network connectivity

As ZebraNet relies on animal movements to create an ad hoc network, how these zebras move and interact critically determines the topology and connectivity of the network, which influences the performance of routing protocols. Hence, before we evaluate the protocols, we first characterize network connectivity. There are two measures of connectivity:

- Direct connectivity: This counts neighbors encountered directly by each node. That is, given a circular radio range of radius r, a collar i is a neighbor of collar j if collar i is within r meters of collar j. This is a good indication of the mobility of nodes and their interactions with each other.
- Indirect connectivity: In addition to direct neighbors, indirect connectivity includes nodes that are reachable via multihop relay through neighbors and neighbors' neighbors. This is a good indication of how peer-to-peer networking will work.

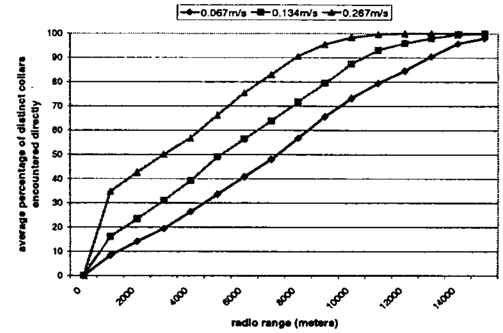


Figure 8: Average percentage of distinct neighbors encountered directly.

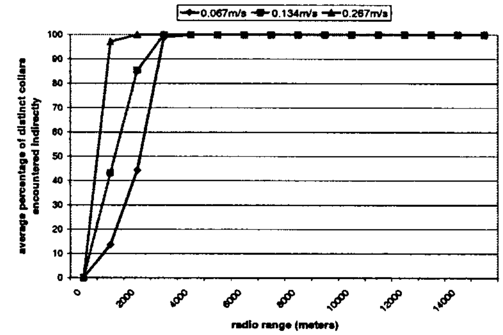


Figure 9: Average percentage of distinct nodes encountered indirectly, through peer-to-peer relaying.

In a mobile ad hoc network, radio range radius r and the mobility of the nodes significantly impact network connectivity. We thus simulate the mobility of zebras at varying r and movement speeds, over a month of simulated movement. Figures 8 and 9 plot the average percentage of distinct nodes zebras encountered directly and indirectly respectively, averaged over the total number of collars. The figures show that as radio range and movement speed increase, direct and indirect network connectivity rise. This is intuitive, since a wider radius r increases the probability of other zebras falling within range. Likewise, a faster-moving animal covers more ground and thus increases the chance of meeting other animals. Figure 8 shows that, using direct neighbors only, 100% connectivity is attained at around 12km radio range for the fastest (0.267m/s) movement speed. If ZebraNet protocols rely solely on direct connectivity to get data back to the base station, they require a very wide 12km radio range that practically covers the entire 20km by 20km map.

Since radio energy consumption increases significantly with radio range (following a square-law or more) a power-efficient network should also tap indirect connectivity through peer-to-peer communication. For the

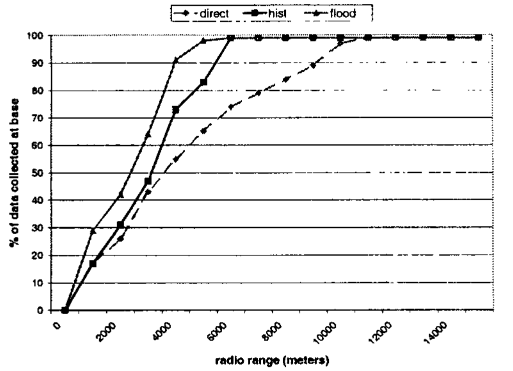


Figure 10: Success rate with infinite storage and bandwidth.

same movement speed of 0.267m/s, Figure 9 shows that using indirect neighbor relationships, the network achieves 100% connectivity with radio ranges of less than 2,000 meters. Hence, peer-to-peer protocols are able to exploit indirect connectivity to reduce radio ranges in sparsely-connected sensor networks, realizing a huge reduction in power consumption. These results support the potential benefits of a peer-to-peer protocol in ZebraNet. They also point to the likely radio ranges we will need to support. The subsections that follow evaluate protocol issues in more detail.

6.3 Protocol Evaluations

To first establish a baseline, Figure 10 shows the success rate of data returned to the base station for an ideal network where there is infinite storage capacity and network bandwidth. We compare three protocols. In addition to the two proposed peer-to-peer protocols—flooding and history-based—we also plot success rate for a protocol that supports no peer-to-peer transfers and only allows direct transmission of a collar's data directly to the base. Both peer-to-peer protocols (flooding and history) perform better than direct transmission, achieving 100% success rate at a radio range of about 6km as compared to 11km radio range needed for 'direct'. This is because the peer-to-peer protocols are better able to percolate data from reclusive nodes that do not meet the base station directly.

We also see that for this unconstrained setup, flooding performs better than the more selective history-based protocol. With no constraints on storage and bandwidth, flooding will have the best performance of any peer-to-peer protocol, since it completely leverages the indirect connectivity of a network, by broadcasting to every neighbor.

6.4 Storage Constraints

As storage capacity is a prominent constraint in the design of sensor nodes, we next investigate the impact of limiting the capacity of onboard memory. To illustrate the trends, we show an extreme case in which storage is limited to 10 collar-days.

As shown in Figure 11, even with storage severely constrained, both peer-to-peer protocols perform better than direct transmission to base. This is somewhat surprising since peer-to-peer requires that the storage handle both the collar's own data as well as that of its peers. The success of the peer-to-peer protocols comes largely due to our deletion strategy which prioritizes a collar's own data over others. This helps ensure that a protocol, at worst, stores only its own data. Hence, the peer-to-peer protocols cannot have a poorer data homing rate than directly transmitting data to base.

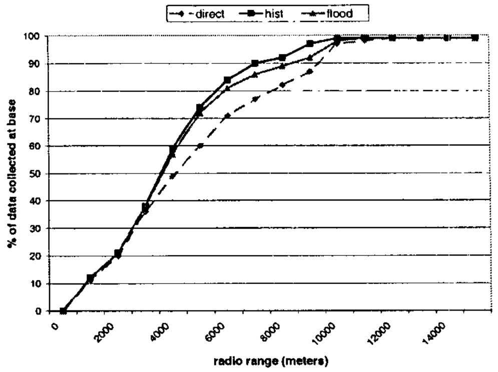


Figure 11: Success rate with constrained storage and infinite bandwidth.

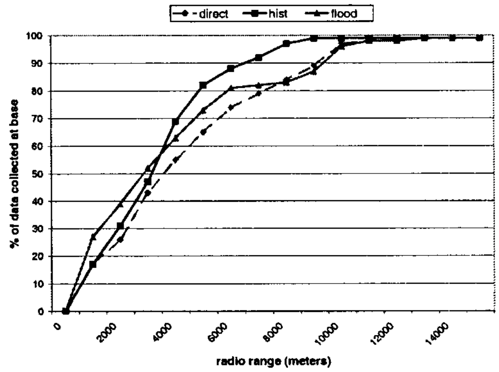


Figure 12: Success rate with infinite storage and constrained bandwidth.

Comparing the 'flood' and 'hist' curves shows that storage contraints degrade flooding's success rate more than that of the history-based protocol. This is fairly intuitive since flooding indiscriminately forwards data to all neighbors, resulting in large duplication of data around the network.

6.5 Bandwidth Constraints

The second major design constraint in sensor networks is bandwidth. Figure 12 shows the success rate of both protocols when the bandwidth is throttled at 12kbps. (To separate the different constraint effects, storage here is once again infinite.) At short radio ranges, below about 4000m, network connectivity is low. As shown in Figure 8 each node sees relatively few neighbors and thus there is relatively little peer data to be transmitted. In this realm, flooding is not yet bandwidth-constrained, so it returns more data than the history-based protocol.

As radio range increases, however, network connectivity rises and the amount of data flooded across the network begins to saturate the available bandwidth. Flooding thus begins to be limited by the 30-minute communications period available with the tight bandwidth-constraint; as a result, its success rate suffers as it blasts redundant data much of the time. The history-based protocol, on the other hand, uses more intelligent selection of which nodes to swap data with, and thereby delivers more useful data to base.

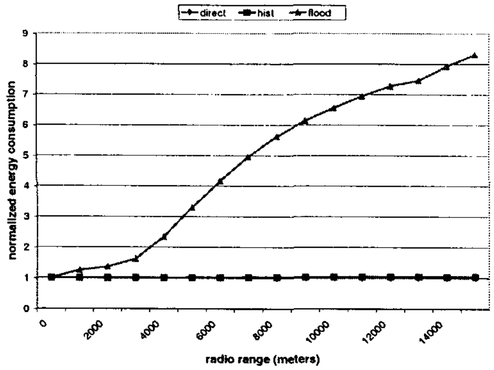


Figure 13: Normalized energy consumption for the non-resource-constrained case. The energy consumed by direct transmission is very close to that dissipated by the history-based protocol.

When running our simulations without prioritizing local collar data over peers, flooding does even worse, as each collar wastes too much time transmitting another collar's likely-redundant data instead of its own data.

6.6 Energy Tradeoffs

Besides success rate, another metric of key interest in sensor networks is the energy consumption. Figure 13 shows the energy consumption for the protocols running on a non-resource-constrained network. The peer-to-peer protocols are shown as energy costs normalized to that of direct transmission, which is plotted as always equal to one. Flooding's energy consumption is more than 8X that of direct transmission at large radio ranges. In constrast, the relative energy of the history-based protocol grows very slowly from 1.0X at 1km radio range to 1.04X at a radio range of 15km. This is expected, since flooding sends messages to everyone in range, when only one copy is needed back at the base. Furthermore, flooding may perform many redundant swaps of data that has already been delivered to the base in cases when the delete-list percolates only very slowly back from the base station. History-based, on the other hand, sends its data to only one receiver. Thus, we see that while flooding typically gives the best performance in peer-to-peer networks with no constraints on storage and bandwidth, its real-life energy cost and bandwidth expectations are exorbitant for large radio ranges. While flooding makes sense at low-radio-range and low-connectivity points in the design space, it is a poor choice for the high-connectivity regime.

6.7 Final Design Choices

The trends summarized in this selection have helped guide the design choices in the ZebraNet prototype node, and in fact, led to our selection of two radios in ZebraNet. The first radio is a low-power, short-range (100m, 19.2Kbps) radio intended mainly for peer-to-peer communications. The second radio is higher-power and longer range (8km at 2.4Kbps) and is intended mainly for transmitting to base. With the uBlox chip providing 640KB of user-accessible flash memory, storage is essentially unconstrained. Simulating a flooding protocol for short range and a direct protocol for long range, our simulations show an 83% success rate, with an estimated 855kJ (66 ampere-hours) energy consumption per month. We are currently experimenting with adaptive protocol variations that should increase the protocol success rate while holding energy roughly constant.

7. RELATED WORK

Sensor networks in general, and environmental sensing in particular, are areas of considerable research interest. This section touches on some of the most salient related work for ZebraNet, divided into sections on environmental applications and wildlife, sensor node design, and protocol studies for sensor networks.

Environmental and Wildlife Sensing Prior wildlife monitoring work for large mammals has almost exclusively been supported by relatively low-technology VHF transceivers that periodically send out a ping signal [10]. More recent improvements have included GPS-based trackers, which have been used for tracking of various animals including birds [27] and sea turtles [4], but these rely on high-power transmitters that transmit data up to a satellite, and they operate off a non-recharged battery supply. Sensor networks have also been proposed for intruder detection, temperature monitoring, and traffic control [7, 36]. Environmental monitoring using sensor nodes with embedded processors are also a focus of the habitat monitoring project [21]. There, they plant the sensors statically in a grid-like fashion across two wildlife habitats. These sensors identify animals when they move through the multiple sensors, and report observed phenomena back to a base station through peer-to-peer transfers through the sensor network. While this has issues in common with ZebraNet, the key difference lies in mobility. In habitat monitoring, sensor nodes are fixed, tracking a dynamic phenomena (moving animal), and reporting to a fixed-location base station. In ZebraNet, sensor nodes and the base station itself are all mobile and only intermittently available for communication. Routing choices thus become more acute for ZebraNet.

Sensor Node Design In the research arena, we bear some resemblance to the TinyOS and TinyNetworkedDevices project [13]. Key differences here are that the Smart Dust "motes" are much more fine-grained than ZebraNet nodes, which include GPS and a 20MHz processor. Thus, they are targeted at different points in the node design space. Ranghunathan et al. have also studied energy and other design issues in sensor networks, discussing different node alternatives [32].

More coarse-grained, the Hiker's Buddy work from Duke looked at power-aware computing issues for a mobile "platform" including a PDA and GPS [6]. This, on the other hand, is actually more coarse-grained in terms of software and energy consumption than what we wish for ZebraNet. It also is intended for direct human use and so did not consider peer-to-peer forwarding of position data to a base station archive.

Protocol Studies Moving more specifically to ZebraNet's communication mechanisms, ZebraNet is a mobile ad hoc network, a research area that has seen increasing attention in recent years. The zebras (nodes) move dynamically and arbitrarily, so the wireless interconnections between the nodes change continually. In a mobile ad-hoc network, the routing protocol has to deliver messages quickly, in the face of unpredictable topology changes. In addition, power efficiency is critical. Numerous routing protocols have been proposed [33]. Some proactively search for routes to all other nodes [26, 29], while others only look for a path when a message needs to be delivered [30, 15]. In ZebraNet, our destination (base station) is only sporadically available. Thus, caching routes (DSR [15]) or significant link state

(AODV [30]) will be ineffective, because cached data may guide data unnecessarily to the base station when it is down. Furthermore, frequent node movements may trigger wasteful cache flushes and route re-discovery. Of the many proposed ad hoc protocols, our ZebraNet protocol most closely resembles epidemic routing [38].

General mobile ad hoc network protocols target arbitrary data flow patterns between multiple sources and destinations. In ZebraNet, data flows either from all the zebras (nodes) to a single destination (base station) *i.e.* data gathering; or occasionally, from a single source to all the nodes, *i.e.* broadcast. These data flow patterns associate ZebraNet closely with the sensor networks subclass of mobile ad-hoc networks [7, 31, 8]. In sensor networks, data is gathered from numerous distributed sensors to a base station, so data too aggregates from many nodes to a single destination. Similarly, the base station broadcasts the information it is interested in to all sensors [14].

The unique communication characteristics of sensor networks have led researchers to study specific routing algorithms for them, since routing protocols proposed for general mobile ad-hoc networks do not work well [8]. However, the routing algorithms proposed thus far [12, 17, 28] assume static sensor networks, i.e., networks where the sensors do not move once they are deployed. Algorithms also assume that the base station stays at a fixed location. Based on the taxonomy proposed in [36], however, ZebraNet is a dynamic sensor network; its nodes are mobile, and so is the base station. This sub-case has not previously been studied in detail.

Another interesting area of research is on connectivity and coverage problems in wireless ad hoc networks. While coverage issues in celluar networks have been well studied, issues of connectivity and "critical mass" for mobile ad hoc networks are still open topics in both theory and systems [11, 23, 24]. Thus far, computational geometry or random graph theory techniques have been applied to global views of network topology. Our work in ZebraNet has focused on stochastic studies based on detailed mobility models. Finally, there has also been work on high level data processing and programming in sensor network to reduce bandwidth or storage needs [3, 14, 20].

8. CONCLUSIONS

This paper discusses the design tradeoffs and early experiences in building a low-power wireless system for position tracking of wildlife. By using peer-to-peer networking techniques, our system can forward data to a researcher's mobile base station without assuming the presence of any cellular phone service or widely-available telecommunications support.

We present initial design ideas, measurements, and weight estimates, and we discuss how battery and weight limits translate into energy and storage limits for our system and its protocols.

Although our protocol development is still very much underway, we feel that the early protocol data the paper provides may be generally useful to the ad hoc networking and systems communities. It represents new steps in protocols for mobile sensor networks, and offers insights into how storage and energy limits may impact protocol design. We are currently making further protocol improvements that will include: (i) position-based, in addition to history-based routing, (ii) self-adaptive decisions on the number of nodes to forward to in the history-based approach, (iii) better support for diverse mobility models. In particular, by having protocols that well-support nodes of disparate speeds, we will be able to collar and study diverse sets of species within the same ecosystem. Finally, we note that our history-based approach currently is stateless (it transfers the information as part of the peer discovery process); we are considering state-based approaches that might decrease peer discovery time.

Overall, ad hoc networking is presently a very active research area. Our work on ZebraNet makes a significant contribution to that domain by offering detailed systems-level perspectives on how to build low-power peer-to-peer systems that operate effectively and are optimized to the characteristics of a particular application domain.

9. ACKNOWLEDGMENTS

We thank Chidozie Enyinna, Ting Liu, Kinari Patel, Karen Tang, and Jeremy Wall for their contributions to the ZebraNet project and for their comments on ideas related to this paper. Julie Buechner's senior thesis project provided the animal observation data in Section 3. We also thank the anonymous reviewers for their comments on the paper. The ZebraNet project is supported in part by an NSF ITR grant. Mpala Research Centre is administered by the Mpala Research Trust, a collaboration of Princeton University, the Smithsonian Institution, the National Museums of Kenya, the Kenya Wildlife Service, and the Mpala Wildlife Foundation.

10. REFERENCES

[1] J. Altmann. Observational study of behavior: sampling methods. *Behavior*, 49:227–267, 1974.

[2] J. Beuchner. Movement rules of african ungulates. Princeton University Senior Undergraduate Thesis. Dept. of Ecology and Evolutionary Biology. Advised by Prof. Dan Rubenstein., June 2002.

[3] P. Bonnet, J. E. Gehrke, and P. Seshadri. Towards Sensor Database Systems. In *Proc. Second International Conference on Mobile Data Management*, Jan. 2001.

[4] Carribean Conservation Corporation. http://www.cccturtle.org/satwelc.htm. 2002.

[5] P. Eggenburger. GPS-MS1E Miniature GPS Receiver Module Data Sheet. u-Blox AG. URL: http://www.u-blox.ch/, Oct. 2001.

[6] C. Ellis. The Case for Higher-level Power Management. In *Proc. HotOS*, Mar. 1999.

[7] D. Estrin, D. Culler, K. Pister, and G. Sukhatme. Connecting the Physical World with Pervasive Networks. In *Pervasive Computing*, Jan. 2002.

[8] D. Estrin, R. Govindran, J. Heidemann, and S. Kumar. Next Century Challenges: Scalable Coordination in Sensor Networks. In *Proc. Fifth ACM/IEEE Int'l Conf. on Mobile Computing and Networking*, Aug. 1999.

[9] D. Ganesan, B. Krishnamurthy, A. Woo, D. Culler, D. Estrin, and S. Wicker. An empirical study of epidemic algorithms in large scale multihop wireless networks. under submission, 2002.

[10] Habit Research. http://www.habitresearch.com/. 2002.

[11] Z. Hass. On the Relaying Capability of the Reconfigurable Wireless Network. In *Proc. IEEE 47th Vehicular Technology Conference, Vol 2, pp. 1148-1152*, May 1997.

[12] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In *Hawaii International Conference on System Sciences*, Jan. 2000.

[13] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister. System Architecture Directions for Networked Sensors. In *Proc. Architectural Support for Programming Languages and Operating Systems*, pages 93–104, 2000.

[14] C. Intanagonwiwat, R. Govindran, and D. Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *Proc. Sixth ACM/IEEE Int'l Conf. on Mobile Computing and Networking*, Aug. 1999.

[15] D. Johnson and D. Maltz. Dynamic Source Routing in Ad-Hoc Wireless Networks. In T. Imielinski and H. Korth, editors, *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996.

[16] P. Karn. MACA - A New Channel Access Method for Packet Radio. In *Proc. 9th Computer Networking Conference*, 1990.

[17] S. Lindsey, C. Raghavendra, and K. Sivalingam. Data Gathering in Sensor Networks using the Energy*Delay Metric. In *Proc. Parallel and Distributed Processing Symposium*, Apr. 2001.

[18] LINX Technologies, Inc. Sc series transceiver module design guide. http://www.linxtechnologies.com/, Apr. 2001.

[19] Lotek Corp. http://www.lotek.com. 2002.

[20] S. Madden, R. Szewczyk, M. J. Franklin, and D. Culler. Supporting Aggregate Queries Over Ad-Hoc Wireless Sensor Networks. In *Proc. 4th IEEE Workshop on Mobile Computing Systems and Applications*, June 2002.

[21] A. Mainwaring, J. Polastre, R. Szewczyk, and D. Culler. Wireless Sensor Networks for Habitat Monitoring. In *Intel Research, IRB-TR-02-006*, June 2002.

[22] Maxstream. 24xstream wireless modem datasheet. http://www.maxstream.net/, July 2002.

[23] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava. Coverage Problems in Wireless Ad-Hoc Sensor Networks. In *Proc. IEEE Infocom 2001, Vol 3, pp. 1380-1387*, Apr. 2001.

[24] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak. Exposure In Wireless Ad Hoc Sensor Networks. In *Proc. MobiCom '01, pp. 139-150*, July 2001.

[25] Mpala Research Centre. http://www.nasm.edu/ceps/mpala. 2002.

[26] S. Murthy and J. Garcia-Luna-Aceves. An Efficient Routing Protocol for Wireless Networks. *ACM Mobile Networks and Applications Journal, Special Issue on Routing in Mobile Communication Networks*, Oct. 1996.

[27] NASA Satellite Tracking of Threatened Species. http://sdcd.gsfc.nasa.gov/ISTO/satellite_tracking/. 2002.

[28] C. Okino and M. Corr. Statistically Accurate Sensor Networking. In *Proc. Wireless Communication and Networking Conference*, Mar. 2002.

[29] C. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. *Computer Communications Review*, Oct. 1994.

[30] C. Perkins and E. Royer. Ad Hoc On Demand Distance Vector (AODV) Routing. In *Proc. IEEE Workshop on Mobile Computing Systems and Applications*, Feb. 1999.

[31] G. Pottie and W. Kaiser. Wireless Integrated Network Sensors. *Communications of the ACM*, May 2000.

[32] Raghunathan et al. Energy-Aware Wireless Microsensor Networks. *IEEE Signal Processing Magazine*, Mar. 2002.

[33] E. Royer and C.-K. Toh. A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks. *IEEE Personal Comunications Magazine*, Apr. 1999.

[34] D. I. Rubenstein. The ecology of female social behavior in horses, zebras, and asses. In P. Jarman and A. Rossiter, editors, *Animal Societies: Individuals, Interactions and Organizations*, pages 13–28. Kyoto University Press, 1994.

[35] Symbol Technologies Inc. S24 High Rate OEM Module Integration Guide. Revision A. http://www.symbol.com, Feb. 2002.

[36] S. Tilak, N. Abu-Ghazaleh, and W. Heinzelman. A Taxonomy of Wireless Micro-Sensor Network Models. June 2002.

[37] United Solar Systems Corp. Web page and technical specifications. http://ovonic.com/unitedsolar/flexiblemodulespecs.html, 2002.

[38] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical report, Apr. 2000. Technical Report CS-200006, Duke University.