Quality of Service and Capacity in Constrained Intermittent-Connectivity Networks

Tara Small, *Member*, *IEEE*, and Zygmunt J. Haas, *Fellow*, *IEEE*

Abstract—In an intermittent-connectivity network, there rarely exists a connected path between a source node and its destination. These networks arise frequently when each node has a limited transmission range, such as a communication network between separated villages or a surveillance network with a large geographical span. One method of addressing the low connectivity of the network uses redundancy. A node generates and stores data; upon reaching the communication range of another node, it replicates the data to it. Multiple copies of the packet decrease the time to offload the data to the destination, but increase the energy and storage used in the system. In this paper, we quantify the resource-delay trade-off and the throughput capacity for intermittent-connectivity networks with Quality of Service restrictions such as limited communication bandwidth. Many routing protocols have been proposed for these intermittent-connectivity networks. Using the *Shared Wireless Infostation Model* as an example strategy, we mathematically represent the intermittent-connectivity network and adjust the model to include a Quality of Service constraint. By completely defining a mathematical model, we allow network designers control over system performance through the adjustment of allocated resources such as communication bandwidth, fraction of time a node spends in sleep mode, or required reliability of packet offloading.

Index Terms—Markov processes, stochastic processes, wireless sensor networks.

1 Introduction

Intermittent-connectivity mobile wireless networks are collections of a relatively small number of wireless and possibly mobile hosts distributed over a large coverage area. Nodes have a small average number of neighbors (typically less than one) and suffer from frequent, often chronic, partitions. A randomly selected pair of nodes is often unable to communicate directly due to large separation between the nodes, power constraints, and signal propagation impairments (such as line-of-sight limitations or propagation obstacles). Almost all traditional routing protocols, wired and wireless [12], assume the existence of connected paths between the message sources and their destinations during the transmission and forwarding[1] of data. We analyze protocols that alleviate the problem of frequently disconnected paths by having a node store the packet, carry it until meeting another relay node (or the destination node), and forward the packet to the other relay node. We term this type of routing the *store-carry-and-forward* paradigm. A packet is considered successfully delivered if a node carrying the packet encounters the destination node and offloads the packet to the destination node. Thus, using the mobility of the nodes themselves, the network successfully delivers packets by forwarding them along *virtual links*, which are created by the movement of network nodes.

1. Thus, the name *store-and-forward* routing

- *T. Small can be reached at 25 Colonial Heights, Fredericton, NB E3B 5M2, Canada. E-mail: tmsmall@gmail.com.*
- *Z.J. Haas is with the School of Electrical Engineering, Cornell University, 323 Frank Rhodes Hall, Ithaca, NY 14853. E-mail: haas@ece.cornell.edu.*

Manuscript received 18 Sept. 2005; accepted 28 Aug. 2006; published online 7 Feb. 2007.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0279-0905.
Digital Object Identifier no. 10.1109/TMC.2007.1033.

The *store-carry-and-forward* networking paradigm and the associated *virtual links* are quite useful in establishing communication in intermittent-connectivity networks. Unfortunately, it is also a fact that this strategy often leads to high latency since nodes can carry the packets for a long time while disconnected from the network. Of course, when network latency is not critical, such as is the case in *delay-tolerant* networks, the *store-carry-and-forward* paradigm can prove to be adequate. For example, this is the case when the delivery of the messages is more important than its delay. With the *store-carry-and-forward* communication paradigm, the delays of packets depend on the rate at which virtual links are created in the network, as well as the availability of network resources, such as storage space and energy.

In our model of an intermittent-connectivity network, each node generates information at some rate λ (in units of [packets/time-step]) to be transmitted to the appropriate destination. In a sensor network, these destinations are designated sinks called collection stations and we assume that they do not change. Examples of intermittent-connectivity networks include:

- networks of animal tags, where the nodes have only limited communication range compared to the distances over which they travel,
- sensor networks that collect statistical information using low-power devices,
- networks of people from separated villages wishing to communicate, and
- hikers in a national park who carry devices to collect information about the trail conditions to be shared with other hikers.

In this work, we derive a model of intermittent-connectivity networks and relate the model's parameters to practical applications. We first introduce a mathematical

Markov chain model for a sparse intermittent-connectivity network and describe the way in which a mobility pattern of the nodes can be expressed as parameters of the model. We then explain the manner in which the model can be used to calculate network performance metrics such as packet delays and the impact of changing parameters of the model.

We then precisely relate parameters of the mathematical model to practical quantities like the communication bandwidth. By using this relationship, a network designer can analytically study the way in which performance metrics change when the physical network resources vary. Furthermore, we extend the above model to include antipackets, a mechanism to reduce storage in the system, and we relate physical quantities to network parameters in this more complex model. To culminate the paper, the relationship between theoretical parameters and physical resources assists us in the calculation of the throughput capacity of an intermittent-connectivity network.

2 Related Work

Grossglauser and Tse [5] showed that the long-term throughput for a source-destination pair can remain constant as the density of nodes increases. In order to provide this constant throughput, messages travel at most two hops: Either the source node reaches the destination directly or the source broadcasts to relay nodes and at least one of these relay nodes reaches the destination. This algorithm is based on the idea that maximizing throughput involves scheduling transmissions over sufficiently short paths at any given time. Since information is not passed until the nodes are sufficiently close to one another, the resource utilization is small. The authors also prove that the message is guaranteed to be delivered in some finite average time (given infinite storage capacity at the nodes), although this time may be very long. This may be a reasonable assumption since intermittent-connectivity networks must inherently tolerate large delays.

An intermittent-connectivity network is an example of a delay-tolerant network, an important type of networks that is gaining interest in the networking community. The Delay Tolerant Networking Research Group (DTNRG) focuses primarily on data delivery in frequently partitioned networks with predictable movement of nodes [2]. The semantics of the US Postal Service provide a basis for the data delivery. Packets attain relative priorities, including custodial delivery (where a message is passed between custodians that become responsible for the reliable transmission of the packet) and return receipts. The authors of [2] define an expiration time for the delivery of a packet as the time at which the message is no longer useful; this functionality is accomplished using local clocks with the current time-of-day. Certain considerations are included, such as security and the ability to save state when a session terminates. Since a node may often reboot or undergo long delays with intermittent connectivity, the ability to begin again in the same state is valuable.

Outside the DTNRG, there have been numerous other works on intermittent-connectivity networking. Proactive approaches for routing in intermittent-connectivity networks involve nodes that modify their trajectories for the purpose of improving communication. Li and Rus [8] proposed an optimal algorithm to compute the trajectories of these nodes so that the message transmission delay is minimized. We do not consider proactive methods in this paper, although we assume that timely delivery is desired and we rely on the natural mobility of the nodes.

Shah et al. [15], [7] introduced a three-tiered architecture to reduce the power requirements of mobile sensors. The work establishes a collection of mobile entities called dataMULEs (Mobile Ubiquitous LAN Extensions), placed on creatures or devices that are already present in the environment. Their scheme uses the natural motion of the nodes and devices to create virtual links in which the dataMULEs approach the sensors, retrieve data, travel, and eventually deliver the data to the collection stations. This scheme provides a low-power transport medium to recover sensor data. Hence, this model is well-suited to sensor networks, where power budgets at the nodes is the biggest constraint, but the sensors' data do not have severe latency restrictions. Shah et al. reduced the space of mobility patterns to a random walk and used stochastic models to examine the success rate of packet delivery and the buffer capacity requirements on sensors and MULEs.

Zhao et al. [21] used special mobile nodes, called "message ferries," to aid communication services, similarly to dataMULEs. However, the ferries follow a nonrandom predetermined route (known by the nodes), rather than relying on their own trajectories. This significantly improves the latency over the dataMULE case. Whenever a source node generates a packet to be delivered to an out-of-range destination node, the source transmits the message to a ferry and the ferry delivers the message. In this way, messages need to travel only two hops, reducing the bandwidth utilization, the contention in the system, and the energy expenditure at the nodes, much like in Grossglauser and Tse [5]. Indeed, performance of the system is considerably improved over other intermittent-connectivity ad hoc networking methods. However, the message delay can still be considerable because the ferry travels primarily on its predetermined route, which passes by nearly all nodes once before returning to any of them.

Epidemic routing [20] involves replication and propagation of copies of a message to many mobile nodes. In epidemic routing algorithms, mobile hosts forward their packets to randomly chosen relay nodes, as well as retaining a copy of the packet. The name stems from the behavior of a message, which is similar to an infectious disease. A node carrying a packet is like a carrier of an infectious disease; we deem such a node both infected and infective. A susceptible node does not carry the packet, but has the potential to carry it. An infective node replicates and forwards a message to a susceptible one once they come into communication range due to their mobility. A basic result of the epidemic theory states that this entire population will be infected in finite average time. Starting from a single infective individual, this is achieved in expected time proportional to the logarithm of the population size [4].

Unfortunately, epidemic routing can introduce a lot of redundant packets, so packets are often dropped due to limitations of local buffer space. In addition, the scheme drains considerable energy to "infect" all of the nodes with the packets. Introducing a maximum hop count partially relieves this inefficient use of resources. For example, a forwarded message with a reduced hop count of 1 can only be offloaded to the destination. Given adequate buffer sizes, this strategy obtains nearly 100 percent message delivery since the packets remain in the system a very long time before they are eventually overwritten by newer packets. Limiting this hop count balances the heavy resource consumption since the message is now copied to fewer nodes, thus exploiting the inherent trade-off between resource consumption and delivery rate/latency.

Davis et al. [3] extended the epidemic routing concept by exploiting nodal movement patterns to forward packets strategically in intermittent-connectivity networks. Using their scheme, nodes learn about the movement patterns of the network nodes when forwarding messages. Additionally, nodes drop packets cleverly when local buffers are full by calculating the probabilities of message delivery based on movement. Lindgren et al. [9] also calculated probabilities of delivery to forward packets, assuming that two nodes are more likely to meet again if they have met recently.

Haas and Small [6] developed the Shared Wireless Infostation Model (SWIM) to extend the epidemic routing concept in a different way. Based on the epidemic theory, the Markov models of SWIM are refined to better and more flexibly represent the networking scenarios. These models include a Time-To-Live (TTL) field that limits resource consumption instead of a maximum hop count. This means that all copies of a packet are erased from the system after TTL time-steps from the time of creation of the original packet (much like in the DTNRG draft [2]). SWIM allows the network designer to choose what fraction of packets to offload by properly setting the value of TTL. For example, the network designer may set the TTL to offload 95 percent of the generated packets (which we label $TTL_{0.95}$). By requiring a smaller fraction of packets to be offloaded, the designer would trade reliability for storage space since $TTL_x < TTL_{x'}$ for $x < x'$. As shown later, the packets remain in the system a much shorter time if they only need to be offloaded with, say, probability of 0.9, rather than a probability closer to 1. Usually, in these types of networks, 100 percent reliability of offloading is unnecessary.

Mobile sensor networks provide an appropriate application for SWIM. Sensor nodes periodically generate data in the network. Then, these nodes function cooperatively to offload the data to collection stations, in fact creating a sensor network [1], [16], [6]. Feedback mechanisms reduce the storage at the nodes by eliminating packets whose copy has already been offloaded and which are no longer necessary. Upon offloading of a packet, a collection station leaves a small identifier at the offloading node, indicating packet delivery. The identifier is spread through the network, so any node carrying a copy of a packet with that identifier can erase the packet, knowing that it is no longer needed in the system.

In this paper, we study the mathematical representation of intermittent-connectivity networks with different types of Quality of Service constraints. Using the analytical results obtainable using this model, a network designer is able to precisely control the cost-benefit ratio, that is, the network resource cost compared to the benefits of system performance. Section 3 defines our model for an intermittent-connectivity network with resource constraints. The model is defined generally enough that many different mobility patterns can be represented by simply recalculating the mobility parameters. The packet delay (latency) is derived for this general model and the implications of more and less severe resource restrictions are discussed in terms of the delay. Section 4 focuses on one particular resource constraint, the communication bandwidth, as a concrete example that limits the intermittent-connectivity system. Simulation results show that we are able to represent the resource constraint in our model well and evaluate the impact of the constraints on the system performance. Section 5 calculates the capacity and utilization of a constrained network. Section 6 concludes the paper.

3 Modeling Systems with Intermittent Connectivity

Consider an intermittent-connectivity network where nodes generate information packets with a Poisson arrival process. In an effort to deliver the packets to their destinations more quickly, nodes copy packets to their in-range neighbors (possibly only to one neighbor, ferry, or dataMULE). This means that multiple copies of each unique packet exist in the system and the packet is offloaded once any of the nodes carrying a copy comes into contact with a collection station. Given a certain mobility pattern, a node may be within communication range of a relay or its destination for a short time only. An *interaction* between two nodes is defined as the time period in which the two nodes are in communication range. Let us assume for simplicity that our network is a sensor network and that the destinations for all of the packets are fixed nodes that we call collection stations. Let L_n be a random variable representing the length of an interaction time between two nodes and L_s be a random variable representing be the length of an interaction between a node and a collection station.

During each interaction between two nodes, each node replicates all of the packets in its queue at the other interacting node. The modeling of a system with multiple different packets is complicated, so we choose to model the replication of one packet at a time. If an unlimited number of packets can be sent in any interaction between two nodes, then this method is clearly acceptable, since the replication of one unique packet will not affect the replication of another packet in any way. However, to represent practical situations, we need to limit the number of packets that can be sent in interactions between nodes. In particular, if we choose a bandwidth constraint b_n (in packets per time-step), then only $b_n l_n$ packets can be sent in a node-to-node interaction of duration l_n, so sending one packet could affect another. As we will soon see, it is still sufficient to model one unique packet and its copies at a time in this case.

In our system, each node has a list of packets, ordered by packet identifiers. At the beginning of an interaction, the

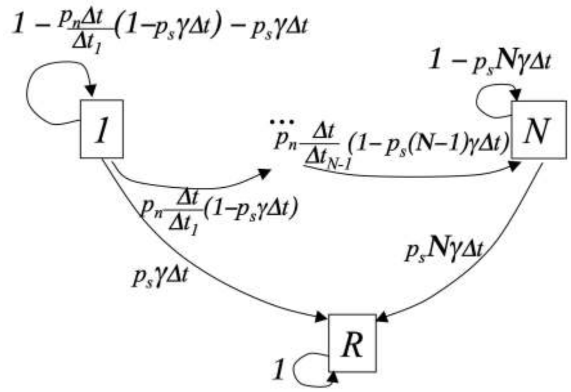

Fig. 1. Markov model for a single packet and its copies.

node chooses a location in the list uniformly and at random. The node sends $b_n l_n$ packets of the queued packets beginning at the random location, possibly starting from the top of the list if the end is reached.[2] By this definition, all of the packets are treated the same way. If $b_n l_n$ of the packets are sent from a queue with length q, then this can be modeled as sending any individual packet with probability $p_n = \frac{b_n l_n}{q}$. In the same way, the bandwidth is limited between a node and a collection station by some different constraint b_s (in packets per time-step). This models an individual packet being offloaded in an interaction of duration l_s with probability $p_s = \frac{b_s l_s}{q}$. We assume that there are no transmission losses in this system; the only loss is due to the expiration of the Time-To-Live of packets.

3.1 Setting Up the Model

Since nodes in intermittent-connectivity networks have less than one neighbor on average and it is sufficient to consider the replications of one unique packet, we can construct a relatively simple mathematical model, the *Shared Wireless Infostation Model* [6]. Note that interference is not a serious issue in these networks due to the scarcity of neighbors. We also assume that the MAC layer is able to schedule packet transmissions such that no collisions occur.

The Markov chain shown in Fig. 1 models the replication in the system for one individual packet. Since it is easier to model a discrete-time system, time is quantized into very small intervals of length Δt which we call "time-steps," allowing only one forwarding event in a single Δt. Each state i represents the number of copies of the packet in the system. R represents the system state in which at least one of the copies has been offloaded to a collection station. $p_s\gamma$ is the rate of offloading from one node to any collection station; this parameter appears in the transitions from nonoffloaded states i to the offloaded state R. If the system is in state i, then i nodes carry copies of the packet so that the rate of offloading is $p_s i\gamma$.

The Δt_i variables represent the average time that the system remains in the state i (i.e., that there are i copies of the packet) before one of the data-carrying nodes is within range of another non-data-carrying node. The packet is replicated with probability p_n so that the transition probability from state i to state $i+1$ is $\frac{p_n \Delta t}{\Delta t_i}$. Since the size of Δt is assumed to be very small, only a single node-to-node forwarding event can occur per time-step; however, during that time-step, it is possible for a copy to be offloaded. The offloading event is given priority in the model since offloading is our primary goal. Therefore, the overall transition probability from state i to state $i+1$ is the conditional probability that the packet is shared between nodes and is not offloaded. This results in the $(1 - p_s i\gamma\Delta t)$ factor. For reference, all of the notations used in this paper are compiled in a table in Appendix A.

The $N+1$ parameters in our model, $\Delta t_i = \Delta t_i(N)$, $1 \leq i \leq N$, and γ, are defined by the mobility pattern. To calculate these parameters, we begin by considering two quantities, the distribution of times until two nodes meet f_X and the distribution of times until one node meets any collection station f_W. If the mobility pattern of the nodes is very complicated, we may need to empirically measure these distributions. However, with reasonably simple mobility patterns, they can be calculated analytically, as shown in Appendix B. We wish to extrapolate what would happen when we add more nodes to the system. Although many different mobility patterns could be used, in this paper, we chose to use a random directional mobility pattern [11] as an example, where the direction is chosen uniformly at random every 15 time-steps and the nodes move linearly.

$\Delta t_1(i+1)$ is the time that it takes one packet-carrying node to reach one of i non-packet-carrying nodes. More precisely, it is the minimum time it takes for the packet-carrying node to reach any of the i non-packet-carrying nodes, where the time to meet any particular non-packet-carrying node has distribution f_X. Therefore, this random variable is $\Delta t_1(i+1) = \min(X_1, \ldots, X_i)$, where $X_j \sim f_X$ are *i.i.d.* $\forall 1 \leq j \leq i$, with the following distribution:

$$\begin{aligned} P(\Delta t_1(i+1) \leq t) &= 1 - P(\text{no interaction by time } t) \\ &= 1 - [1 - P(X \leq t)]^i. \end{aligned} \tag{1}$$

Using the same ideas, we state that

$$\Delta t_i(N) = \min(Y_1, \ldots, Y_i),$$

where $Y_j \sim f_{\Delta t_i(N-i+1)}$ are *i.i.d.* $\forall 1 \leq j \leq i$, because we have i packet-carrying nodes that could each potentially interact with $N-i$ non-packet-carrying nodes. So, $\Delta t_i(N)$ are the Δt_i variables used in the model of Fig. 1.

These models have been shown to work well by taking $\frac{1}{\gamma}$ as the mean of the f_W distribution and using the $\Delta t_i(N)$ calculated above [17]. The probability of one node offloading a packet is $p_s\gamma\Delta t$, and the probability of one of the i nodes offloading a packet is $p_s i\gamma\Delta t$. However, further calculations may be needed if more complicated mobility pattern examples are used. Using the same technique as above, we would let $\frac{1}{\gamma_i}$ be the time that is takes i packet-carrying nodes to reach a stationary collection station. That is, $\frac{1}{\gamma_i} = \min(W_1, \ldots, W_i)$, where $W_j \sim f_W$ are *i.i.d.*, $\forall 1 \leq j \leq i$.

Note that one of the greatest advantages of this model is the generalization of the mobility pattern used. By performing the above calculations on the contact rates to find the model parameters, we can use realistic and potentially complicated mobility patterns that are applicable in

2. The assumption of picking a random location in the queue for transmission is a simplification. In a practical scenario, other scheduling methods, such as first-come-first-serve, could be used.

practical settings. The model of Fig. 1 assumes that all nodes use the same mobility pattern; however, our work can be extended in a straightforward manner to situations where sets of nodes use different movement.

3.2 Calculating the Packet Delay

Using the model in Fig. 1 and appropriate initial conditions, we are able to find the probability distribution, as a function of time, of the different states of the system. We define the random variable T as the delay of a packet, the time from the packet creation (time $t = 0$) until any copy is offloaded to a collection station (i.e., when the system enters state R). Therefore, finding the probability of the system in state R as a function of time provides us with the cumulative distribution function for the delays of the packets, $F(T)$.

Using the cumulative distribution of the delay, we are able to determine the threshold probability P_{thresh} that represents the confidence with which we desire each packet to be successfully offloaded to a collection station. For example, if $P_{\text{thresh}} = 0.5$ and $F^{-1}(0.5) = 200$, then the packets need to remain in the system for 200 time-steps, after which time they are successfully offloaded with probability 0.5 and all of the copies may be removed from the system. This is accomplished by adding a Time-To-Live (TTL) field (of 200, in the above example) to each packet, which is decreased at each time-step. When the $TTL = 0$, the packet is erased. Note that only the remaining TTL time is passed when the packet is shared between nodes; i.e., there is no need for clock synchronization among the nodes. We assume in this section that the expiration of the TTL is the only method by which packets are erased.

Let us assume that the initial probability of state i is a_i and that the initial probability of state R is a_R. Clearly, $a_1 + a_2 + \ldots + a_N + a_R = 1$. Let us consider probabilities $P(\text{state}, \text{time})$.

In the following calculations, let

$$d_i = \frac{p_n \Delta t}{\Delta t_i}(1 - i p_s \gamma \Delta t).$$

First, we calculate the probability of state 1 at any time-step j.

$$P(1, j\Delta t) = a_1(1 - d_1 - p_s \gamma \Delta t)^j. \tag{2}$$

With $P(1, j\Delta t)$ known, we calculate the probabilities of state 2 as

$$\begin{aligned} P(2,0) &= a_2, \\ P(2, j\Delta t) &= P(2,(j-1)\Delta t)(1 - d_2 - 2p_s\gamma\Delta t) \\ &\quad + d_1 P(1,(j-1)\Delta t). \end{aligned} \tag{3}$$

Similarly, we find the probabilities of states i up to $N-1$,

$$\begin{aligned} P(i,0) &= a_i, \\ P(i, j\Delta t) &= P(i,(j-1)\Delta t)(1 - d_i - ip_s\gamma\Delta t) \\ &\quad + d_{i-1} P(i-1,(j-1)\Delta t), \end{aligned} \tag{4}$$

and, for $i = N$,

$$\begin{aligned} P(N,0) &= a_N, \\ P(N, j\Delta t) &= P(N,(j-1)\Delta t)(1 - Np_s\gamma\Delta t) \\ &\quad + d_{N-1} P(N-1,(j-1)\Delta t). \end{aligned} \tag{5}$$

Finally, we find the cumulative distribution of the off-loading times, $F(j\Delta t) = P(R, j\Delta t)$,

$$\begin{aligned} P(R,0) &= a_R, \\ P(R, j\Delta t) &= P(R,(j-1)\Delta t) \\ &\quad + \sum_{i=1}^{N} (ip_s\gamma\Delta t) P(i,(j-1)\Delta t). \end{aligned} \tag{6}$$

3.3 Effect of p_n on Performance

Estimation of energy consumption in the network depends considerably on the energy model used. We assume that each node can have two energy states, *active* (either transmitting a packet, receiving a packet, or sensing the channel in idle mode while waiting for the arrival of new data) or *sleep* (conserving power and unable to transmit or receive any information). Pearlman et al. [13] observed that the transceiver consumes the most energy when transmitting. Receiving and idly listening to the channel both consume less, but comparable power. The power consumption of a node in the idle state is often more than half the power consumed when it is actively transmitting packets. Only the *sleep* state conserves significant energy at the nodes. Stemm and Katz [19] showed through measurements that most of the power drawn from the batteries in a CSMA/CA scheme is due to the sensing mechanism in the idle mode.

If the nodes need to continually sense the channel to determine whether other nodes or collection stations are within range, then the energy dissipation will be similar, regardless of the number of transmissions from the nodes. Let us consider the ideal case where the nodes are able to enter the *sleep* mode most of the time, but are able to transmit/receive at every possible interaction opportunities; the packet transmissions and receptions are the primary source of energy consumption. Our energy metric is the number of times that copies of the packet are transmitted between nodes or from a node to a collection station. Suppose that nodes are put into sleep mode $(1 - p_n)$ fraction of the time and can therefore receive p_n fraction of the time. The collection stations have a renewable energy source so that they can receive at any time and $p_s = 1$. Note that choosing $p_n < 1$ reduces the energy, but does not bound the energy spent per packet. We further investigate bounded energy per packet in [18].

Taking parameter $p_n = 0$ corresponds to the case where no node-to-node copying occurs. This leads to low storage and low energy usage for each unique packet. However, since the single copy of each packet must be transmitted directly to the destination, the delays of packets can be quite long. When p_n increases, packets are copied to other nodes in interactions. More copies of each unique packet lead to a greater chance of offloading one of them, resulting in smaller delays, but more storage space and energy is required per packet.

We exhibit the energy-delay trade-off for a unique packet in Fig. 2 for a system with 40 nodes and one collection station moving in a 150 units by 150 units network area. The communication radii of the nodes and stations are 7 units. This example uses the SWIM strategy to forward and offload packets with a feedback mechanism called VACCINE (Section 4.2). VACCINE reduces the

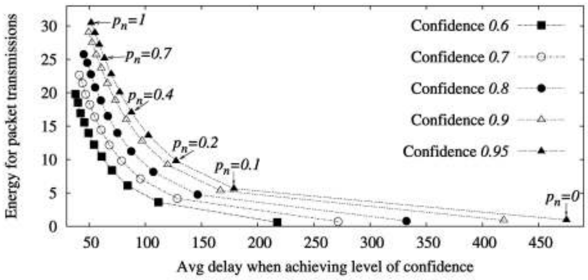

Fig. 2. Energy-delay trade-off.

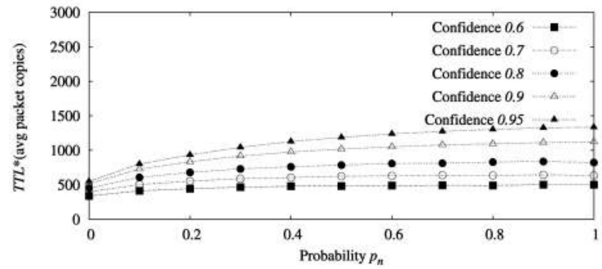

Fig. 3. Indication of total number of packets, $E(N)$, since $E(N)/\lambda = TTL^*$(copies of a unique packet).

storage requirement at nodes by eliminating offloaded packets from the system. Fig. 2 shows large reductions in the delays of the packets for smaller increases in energy for small p_n values, such as $p_n = 0.1$ or $p_n = 0.2$. An energy-conscious network designer may choose to operate the network in this range.

Another important resource is the total storage in the system. By Little's Formula, the expected number of packets in the system is $E(N) = \lambda * TTL *$(average copies of each unique packet in $[0, TTL]$), where λ is the generation rate of the packets. Since λ is constant as we vary p_n, we can consider the product $TTL *$ (copies of each unique packet) for different levels of confidence as a normalized expected number of packets in the system. Recall that, as p_n increases, TTL decreases and the number of copies of each unique packet increases, so it is difficult to predict whether the product increases or decreases. Fig. 3 shows that the expected number of packets in the system increases somewhat with p_n for several levels of confidence in delivery for the chosen set of parameters.

4 Bandwidth-Delay Trade-Off

Having set up all of the mobility parameters in the model shown in Fig. 1, we further investigate the relationship between the probability of sending a packet in an interaction and the corresponding allocated communication bandwidth. We solve this problem in a manner that may seem "backward"[3] by choosing values for p_n and p_s and then solving for the corresponding values of b_n and b_s for a given choice of desired fraction of offloaded packets.

Suppose we choose p_n, p_s, and P_{thresh}. Using the model from Fig. 1, we can find the Time-To-Live of a packet that corresponds to a desired probability of offloading P_{thresh}. As long as $p_s > 0$, a TTL value exists for any P_{thresh}, but it may be very large. In such a case, the queues could get very long, but, since we assume the storage buffers are unbounded, packets are not lost.

Using this TTL value and the model of Fig. 1, we find the average number of copies of a unique packet in the system, $E(I)$, for any time within $[0, TTL]$. If each of the N nodes generates packets by a Poisson process at rate λ, then the distribution of the total packets in the system is approximately

$$P(\text{total packets} = N \cdot E(I) \cdot k) = \frac{e^{-(\lambda \cdot TTL)}(\lambda \cdot TTL)^k}{k!}$$

3. If instead, the b_n and b_s are given, it is difficult to estimate the queue lengths to solve for p_n and p_s.

and, since all nodes are equally likely to carry packets, we can divide the packets among the N nodes and estimate the distribution of packets per node, Q, as

$$P(Q = E(I) \cdot k) = \frac{e^{-(\lambda \cdot TTL)}(\lambda \cdot TTL)^k}{k!}. \quad (7)$$

To find the corresponding b_n and b_s, we also need the distributions of the interaction durations. Recall that L_n is the interaction duration between nodes and L_s is the interaction time between nodes and collection stations. If the mobility pattern is complicated, the distributions of L_n and L_s can be calculated using precise empirical measurements. However, for simple mobility patterns, it can be calculated analytically, as shown for the mobility parameters of the model in Appendix B.

Note that the queue length Q is independent of L_n. Q is calculated using the generation rate λ and the model from Fig. 1, whose parameters do not depend on interaction times.

Equation (8) relates p_n and b_n by mathematically expressing p_n as the fraction of packets sent in a node-to-node interaction averaged over all possible queue lengths and all possible interaction lengths. For each instance of an interaction with a sender queue of size q, the entire queue can be sent if $q \le l_n b_n$, but only $l_n b_n$ of the queue can be sent if $q > l_n b_n$. Since b_n is the only unknown in this formula, we can solve for b_n and discover the $p_n - b_n$ relationship,

$$p_n = \frac{x_n + y_n}{\sum_{l_n=1}^{\infty} \sum_{q=0}^{\infty} qP(Q = q)P(L_n = l_n)}, \quad (8)$$

where $x_n = \sum_{l_n=1}^{\infty} \sum_{q=0}^{l_n b_n} qP(Q = q)P(L_n = l_n)$ and

$$y_n = \sum_{l_n=1}^{\infty} l_n b_n \sum_{q=l_n b_n+1}^{\infty} P(Q = q)P(L_n = l_n).$$

In a similar way, we can say that

$$p_s = \frac{x_s + y_s}{\sum_{l_s=1}^{\infty} \sum_{q=0}^{\infty} qP(Q = q)P(L_s = l_s)}, \quad (9)$$

where $x_s = \sum_{l_s=1}^{\infty} \sum_{q=0}^{l_s b_s} qP(Q = q)P(L_s = l_s)$ and

$$y_s = \sum_{l_s=1}^{\infty} l_s b_s \sum_{q=l_s b_s+1}^{\infty} P(Q = q)P(L_s = l_s).$$

4.1 $p_n - b_n$ Evaluation

Let us examine the calculations of the previous section using a network area of size 300 units by 300 units with 40 nodes and one collection station. The nodes move with

TABLE 1
p_n versus b_n in [Packets/Time-Step] with $p_s = 1$ and $P_{\text{thresh}} = 0.9$

p_n	b_n	p_n	b_n
0	0	0.6	66.4
0.1	12.53	0.7	82.15
0.2	23.59	0.8	105.05
0.3	33.64	0.9	148.65
0.4	43.48	1	540
0.5	54.0		

the following mobility model (random directional mobility). The nodes choose their directions uniformly at random and choose a speed uniformly at random between 0 and 6 units/time-step, move in that direction for 15 time-steps, then choose a new direction without pausing. The communication ranges of the nodes and of the collection station is 7 units.

If we suppose that the stationary collection station can have a large antenna and significant decoding capability (both longer than the mobile nodes), we can assume that $p_s \approx 1$ and find b_n for different values of p_n. These results are shown in Table 1.

To evaluate the accuracy of this relationship, we can use many different metrics. One such metric is the delay, the time from creation of a packet until its reception at a collection station. In Fig. 4a, the analytical cumulative delay distribution is obtained using the model from Fig. 1 with $p_n = 0.4$, 0.7, and 1 and with $p_s = 1$. We are able to estimate the queue lengths for models with multiple packets and find the corresponding values for b_n. The empirical curve uses simulation with TTL corresponding to a probability of offloading of 0.9 and the b_n values with $\lambda = \frac{1}{30}$. The empirical simulation only reports the delays for packets that are offloaded to a collection station and the other packets are lost; therefore, we scale the empirical delay distribution by the fraction of the generated packets that successfully offloaded at least one copy. We can also compare the predicted lengths of the queues to the queue lengths observed in simulation. As shown in Fig. 4b, the queue lengths also agree well.

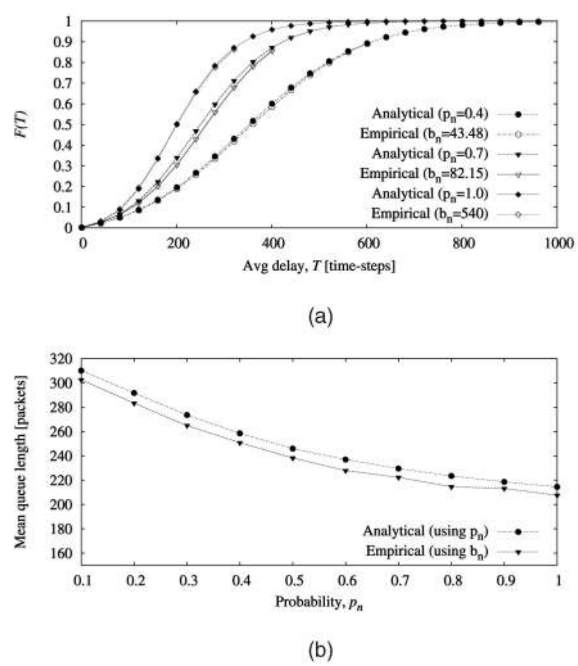

Fig. 4. Comparing theoretical to empirical metrics with $P_{\text{thresh}} = 0.9$. (a) Cumulative delay distributions. (b) Node queues.

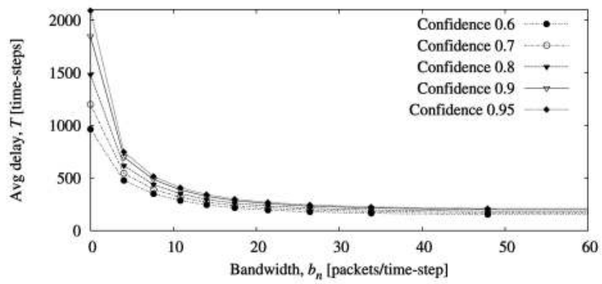

Fig. 5. Average delay versus bandwidth in a 40-node system with one collection station.

With the $p_n - b_n$ relationship, we are able to use our analytical model to examine the performance of a system as the resource constraints vary. Fig. 5 shows how the average packet delay changes in a network where different node-to-node bandwidths are used. Five different levels of confidence in delivery are shown. At low values of node-to-node bandwidth, the replication of packet copies is slower, which leads to longer times that the packet copies must remain in the system to achieve the same offloading probability. At these values of b_n, only a small number of packets are replicated during an interaction. Increasing b_n by a small amount leads to several times the number of packets sent between nodes. This results in a sharp decrease in TTL for small b_n values. However, at large values of b_n, we see little variation in the TTL values required to achieve a particular confidence level of offloading. Increasing the already large b_n values does not reduce the TTL because the sparsity in the network and the rate at which the nodes interact are the bottleneck that limits the system performance.

4.2 Incorporating Antipackets

In order to reduce the storage requirement at the nodes, we introduce *antipackets,* which erase unnecessary packets from the queues of the network nodes [6]. An antipacket is generated when a packet is offloaded to its destination and is given the same identifier as the offloaded packet. These antipackets are transmitted between nodes as they interact, although antipackets are much smaller, so we assume that all of the antipackets in a queue can be sent during a time equal to a single packet transmission time. The purpose of the antipacket is to inform a node that a packet that it holds has already been offloaded. Therefore, the packet is no longer needed and the node can erase that packet from its memory without harming the data delivery in the system. The antipacket carries the remaining Time-To-Live field from the original packet, allowing the nodes to purge all packets and antipackets at the same time. Note that there is no reason to send antipackets in node-to-station interactions; thus, the calculations of the previous section suffice to relate p_s and b_s in this case as well.

Each node carries two queues: one for packets (with queue of size q_p) and one for antipackets (with queue of size q_a). Let Q_p and Q_a be random variables representing the numbers of packets and antipackets in the packet and the antipacket queues, respectively. p_p is the probability of sending a packet (given that $|Q_p| \neq 0$) and p_a is the probability of sending an antipacket (given that $|Q_a| \neq 0$) in a single interaction between nodes. Much like the evaluation of the $p_n - b_n$ relationship of the previous section, the goal in intermittent-connectivity networks with antipackets is to calculate and evaluate the relationship between p_p, p_a, and b_n.

Since each node carries two queues, there is a choice of which queue to send first. The two orderings lead to similar results, so we assume that the antipacket queue is sent first. Recall that all antipackets can be sent in one packet transmission time. If $l_n b_n \geq 1$ for a particular interaction of length l_n, then the antipackets are sent first and up to $(l_n b_n) - 1$ packets are sent in the remaining time. However, if there are no antipackets in the node's queue, then up to $l_n b_n$ packets can be sent. From these descriptions, we derive equations for p_a and p_p, the respective average probabilities of sending an antipacket or a packet in a node-to-node interaction. Recall that all packets (and, therefore, packets and antipackets that have different identifiers) are independent.

Let us assume that the node-to-node bandwidth is large enough that at least one packet can be sent in any interaction; that is, $l_n b_n \geq 1$ and, therefore, $p_a = 1$. As in the previous section, we choose p_p, p_s, and P_{thresh}. We adjust the model from Fig. 1, as shown by Haas and Small [6], and find the Time-To-Live TTL that corresponds to P_{thresh}. Again, we predict the size of the queues for the packets using the expected number of packet copies $E(I)$ using (7). In cases using antipackets, relating these p_p and p_s to the bandwidth constraints requires us to calculate the probability that antipackets exist at the node. This tells us if we can send $l_n b_n$ packets during the interaction or only $(l_n b_n - 1)$ packets.

A node has no antipackets if both of the following independent conditions are true:

- It has no knowledge of any of its current (nonexpired) packets being offloaded to a collection station and
- no antipackets have been received for packets generated at other nodes.

Suppose that packets are independent as before; then, we approximate the probability that none of the node's own packets have created an antipacket, i.e., have been offloaded to a collection station. Using the model of a unique packet and its antipackets, we know that the offloading rate is $p_s\gamma$. Packets are generated reasonably often at the nodes, so we can assume that at least one packet is generated between each two node-to-station interactions. The node still holds antipackets for the remainder of the TTL value after reaching the collection station, so we expect the nodes to hold antipackets that they generate themselves for $p_s\gamma * TTL$ time duration. Also, recall that times between collection station visits are memoryless (exponential), which means that the overall process of visiting a collection station is Poisson. We can then calculate

$$P(\text{no antipackets from node's own offloads}) = \frac{e^{-(p_s\gamma * TTL)}(p_s\gamma * TTL)^0}{0!} = e^{-(p_s\gamma * TTL)}.$$

Next, consider the probability of receiving antipackets from other nodes' offloads. Suppose that we are considering a particular node, call it Node 1. Interactions between Node 1 and another node will occur at the rate $\frac{1}{\Delta t_1(N)}$ in a system with N nodes. If Node 1 receives a set of antipackets from another node, then the "youngest" antipacket could last up to TTL more time-steps. Unfortunately, we do not know the probabilities that the nodes involved in these interactions have antipackets because this is precisely the probability we wish to obtain. We call this probability $P(|Q_a| = 0)$.

$$P(\text{no antipackets from other nodes' offloads}) = \frac{e^{-\left[P(|Q_a|=0)\frac{TTL}{\Delta t_1(N)}\right]}\left(P(|Q_a| = 0)\frac{TTL}{\Delta t_1(N)}\right)^0}{0!} = e^{-\left[\frac{P(|Q_a|=0)TTL}{\Delta t_1(N)}\right]}.$$

We find the overall probability of a node having antipackets by solving

$$P(|Q_a| = 0) = e^{-(p_s\gamma * TTL)} * e^{-\left[\frac{P(|Q_a|=0)*TTL}{\Delta t_1(N)}\right]}. \tag{10}$$

As we conservatively estimated the length of the TTL for the antipackets, this calculated $P(|Q_a| = 0)$ may be slightly lower than the actual value.

Now, we are able to relate the probability p_p to the node-to-node bandwidth.

$$p_p = \frac{(x + y)P(|Q_a| = 0) + (u + v)P(|Q_a| \neq 0)}{\sum_{l_n=1}^{\infty}\sum_{q_p=0}^{\infty} q_p P(Q_p = q_p)P(L_n = l_n)}, \tag{11}$$

where

$$x = \sum_{l_n=1}^{\infty}\sum_{q_p=0}^{l_n b_n} q_p P(Q_p = q_p)P(L_n = l_n),$$
$$y = \sum_{l_n=1}^{\infty} l_n b_n \sum_{q_p=l_n b_n+1}^{\infty} P(Q_p = q_p)P(L_n = l_n),$$
$$u = \sum_{l_n=1}^{\infty}\sum_{q_p=0}^{l_n b_n-1} q_p P(Q_p = q_p)P(L_n = l_n),$$
$$v = \sum_{l_n=1}^{\infty}(l_n b_n - 1)\sum_{q_p=l_n b_n}^{\infty} P(Q_p = q_p) * P(L_n = l_n).$$

To gain more confidence in the accuracy of this relationship, we compare the analytical and the empirical packet delays using the SWIM routing scheme [6] that incorporates antipackets, called VACCINE. The VACCINE method of packet removal generates antipackets at the time of offloading of any packet. These antipackets are passed during a node-to-node interaction, regardless of whether the receiving node has ever stored a copy of that packet before. If the receiver has never stored a copy of the packet, we are "vaccinating" the node such that a copy of the packet will never be accepted. If the receiver has a copy of the packet, it is erased and the antipacket is retained. Fig. 6 shows good

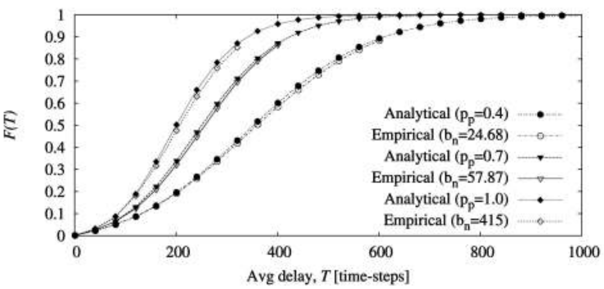

Fig. 6. Comparing analytical and empirical delay metrics with $P_{\text{thresh}} = 0.9$.

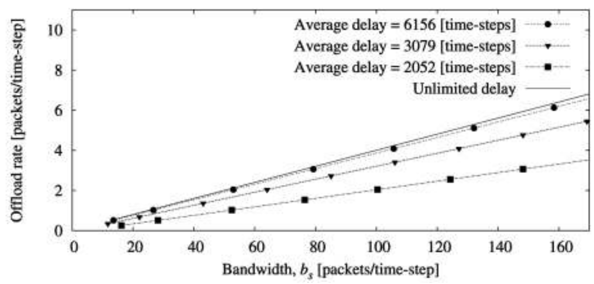

Fig. 7. Capacity that can be achieved with limitations on desired average packet delay.

agreement between the analytical and the empirical delays for this example.

5 On the Capacity Modeling

Using a SWIM system with collection station bandwidth b_s, the best possible offloading rate that can be obtained would offload new data at every possible opportunity. This is achieved if the packet generation rate λ is very large, so that the queues always contain new data to offload at rate b_s for the entire time they are within range of the collection station. The capacity is the maximum offloading rate

$$\begin{aligned} & b_s * \frac{\text{time in communication range of a station}}{\text{total time}} \\ & \quad = b_s * N \frac{E(L_s)}{1/\gamma + E(L_s)} \end{aligned} \tag{12}$$

for a system with N nodes. Equation (12) assumes that either the system is sparse enough[4] or that the station has multiple receivers.[5] This capacity can be achieved with any desired packet offloading fraction P_{thresh} for any bandwidth b_s if the delays and the storage queues can be arbitrarily long.

If we wish to limit the average delay of packets or the average queue size, we can use the theory previously discussed in this paper. Suppose that $0 \leq P_{\text{thresh}} < 1$ is the fraction of packets that we desire to offload successfully. With infinite storage buffers, we can find the finite TTL values that achieve the fraction P_{thresh} of successfully offloaded packets. Expiring packets after TTL means that the queues will be stable (not infinite), but could be very large. As we have seen in the previous sections, we begin by choosing p_n and p_s, finding the queues using the model, and solving for the corresponding b_n and b_s for a particular value of λ. The choice of p_n, p_s, P_{thresh}, and λ completely specify the queue lengths, the Time-To-Live, and average delay for the packets. Therefore, we can calculate the throughput capacity of the channel with the desired average delay.

If the nodes send all of their packets before the end of a node-to-station interaction, then some of the communication bandwidth is wasted; packets could have been sent, but there were no new packets available. The utilization is the fraction of the bandwidth that is used to send packets and is not wasted. Let U_s be the utilization of the bandwidth at the collection stations as nodes are offloading packets.

$$U_s = \frac{u_1 + u_2}{\sum_{l=1}^{\infty} l b_s \sum_{q=0}^{\infty} P(Q = q) P(L_s = l)}, \tag{13}$$

where $u_1 = \sum_{l=1}^{\infty} \sum_{q=0}^{l b_s} q P(Q = q) P(L_s = l)$ and

$$u_2 = \sum_{l=1}^{\infty} l b_s \sum_{q = l b_s + 1}^{\infty} P(Q = q) P(L_s = l).$$

The offloading rate for the unique packets in a system depends on λ, P_{thresh}, b_n, and b_s and can be expressed as

$$\text{offloading rate} = \frac{1}{E(I)} \left[U_s * b_s * N \frac{E(L_s)}{1/\gamma + E(L_s)} \right] \tag{14}$$

if each unique packet has $E(I)$ copies on average. To maximize the offloading rate and to achieve the capacity, our system takes $E(I) = 1$, obtained by setting $p_n = 0$. In this way, there is no redundancy in the offloading to collection stations.

We set $p_n = 0$ and choose some p_s. This fixes the model and the average delay experienced by the packets. Using different packet generation rates λ, we obtain different corresponding values of b_s using (8). Substituting into (13), we calculate the utilization experienced by the system for each choice of λ and, in turn, the packet offloading rate.

The offloading rates in Fig. 7 use a system with 40 nodes and 1 collection station in a toroidal area of size 300 units by 300 units for communication radii of 7 units for both the nodes and the collection stations and a random directional mobility pattern. The "Unlimited delay" curve corresponds to the optimal capacity of the system described by (12). In this case, nodes always have packets to send during node-to-station interactions, so the utilization of the communication channel is $U_s = 1$. However, this also means that there are very long queues and it would take many interactions to send a particular data packet on average.

The other curves of Fig. 7 limit the desired average delay of packets. Mathematically, this is accomplished by increasing p_s. In particular, $p_s = 0.3$, 0.6, and 0.9 are used in the figure. Intuitively, limiting the average delay means that the packets need to be offloaded in fewer interactions on average; the queues must be shorter, so the utilization and overall offloading rate are smaller.

4. That there is at most one node within the range of a collection station at any time.
5. That all the node's messages can be received at once.

TABLE 2
Table of Notations Used in This Paper

Notation	Description
N	total number of nodes in the network
TTL	Time-to-Live of each packet, then the packet is erased
γ (γ_i)	contact rate between one node and the collection stations with N nodes in the system (with i nodes in the system)
Δt_i ($\Delta t_i(j)$)	time expected before i nodes carrying packet copies meet a node without a copy of the packet in a system with N nodes (in a system with j nodes)
P_{thresh}	average probability that a copy of a packet will be successfully received at a collection station
$F(T)$	cumulative distribution function of the packet delays
L_n	random variable representing the time that nodes are within interaction range of each other
L_s	random variable representing the time a node is within interaction range of a collection station
λ	rate of packet generation
$E(I)$	expected number of copies of a packet
Q	random variable representing the length of the packet queue at a node
Q_p	random variable representing the length of the packet queue at a node in a system with antipackets
Q_a	random variable representing the length of the antipacket queue at a node in a system with antipackets
p_n	probability a packet is successfully received in a node-to-node interaction
p_s	probability a packet is successfully received in a node-to-station interaction
p_p	probability a packet is successfully received in a node-to-node interaction with antipackets
p_a	probability an antipacket is successfully received in a node-to-node interaction with antipackets

6 Conclusion

This study analytically quantifies trade-offs between packet delay and network resources by defining a mathematical model that accounts for Quality of Service constraints. Although many constraints could have been considered, we chose a particular QoS constraint in this paper, finite communication bandwidth, to exemplify methods of translating a physical resource constraint into variables of the model. We saw that, when bandwidth is scarce, small increases in bandwidth resources can lead to a considerable reduction in average delay or increase in the fraction of offloaded packets. With moderate to substantial resources available, the performance will improve only slightly because the lack of connectivity, rather than communication resources, is the limiting factor of packet delivery in the network.

The performance of resource-constrained networks is improved by introducing *antipackets* into the system. Antipackets allow nodes to share packets that have not previously been offloaded more frequently because redundant packets can be erased before their Time-To-Live expires. Therefore, the average delays of packets are reduced compared to an equivalent system without antipackets.

The throughput capacity of the intermittent-connectivity networks increases linearly with respect to the available communication bandwidth between the nodes and the collection stations. We are able to set network parameters to achieve different average delays for the packets. However, smaller average packet delays correspond to a smaller rate of increase of throughput capacity.

TABLE 3
Further Notations

Notation	Description
b_n	bandwidth that can be used to send packets between nodes
b_s	bandwidth that can be used to send packets between a node and a collection station
U_s	random variable representing the utilization of the channel between a node and a collection station in a node-to-station interaction

Our analytical model gives network designers control over intermittent-connectivity systems. By allowing the designers to choose the most important parameters to constrain, the model effectively predicts the system performance. One can experiment with expected packet delays for different allocations of network resources and can make educated decisions about the resource allocation in the system. Furthermore, the designer can set the desired fraction of packets to be offloaded and calculate the average output rate. In our future work, we plan to further explore resource conservation by exploiting nonrandomness properties of the network with mobility patterns that better represent real-world scenarios.

Appendix A
Explanation of Notations

See Table 2 and Table 3.

Appendix B
Time Until Two Nodes Meet

For relatively simple mobility patterns such as fluid mobility, we can analytically calculate the time until two nodes meet in a system with only two nodes. Nodes using the fluid mobility pattern choose a direction uniformly at random and continue in that direction indefinitely. Our network area is toroidal, so a node that reaches the boundary on the left (top) side simply reenters on the right (bottom) with the same direction and the same velocity.

At time $t = 0$, a node is placed uniformly at random in a network area with one collection station. Let the total network area have size xy and the initial node position be (x_0, y_0). Without loss of generality (due to the toroidal nature of the area), we assume that the collection station is at the center of the network area. Each time a node crosses a boundary and reenters from the "other side," we can imagine that a new copy has entered the network area. In this way, we form an infinite tiling of network areas and the collection stations appear to be a lattice with vertical separation y and horizontal separation x, as shown in Fig. 8.

We wish to calculate the time for the node to meet any one of the collection station copies. We can think about the distribution of the meeting times as geometric distribution. Each time the node path crosses a copy of the network area, it is considered as one attempt. For each attempt, there is some probability β that the node will successfully reach a collection station.

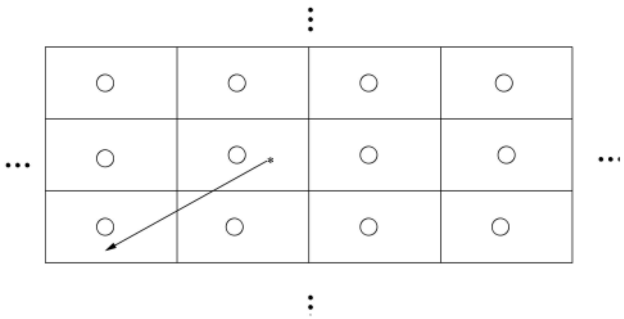

Fig. 8. Node moves with fluid mobility; torus is viewed as many copies of the same area.

In order to simplify the calculations, we suppose that the rectangular network area regions are approximately the same as circular areas with radius $R = \frac{x+y}{4}$, half of the average side length. Assume that the node travels at some constant velocity v. We find X_v, the distribution of times for one attempt when the node is known to be traveling at speed v. If the node begins at the boundary of the circle with radius R at a uniformly randomly chosen entrance angle θ, then, from Fig. 9, we see that the distribution of distances across the circle is

$$\begin{aligned} P(\text{path distance} \leq l) &= P\left[\theta > \cos^{-1}\left(\frac{l}{2R}\right)\right] \\ &= 1 - P\left[\theta \leq \cos^{-1}\left(\frac{l}{2R}\right)\right] \\ &= 1 - \frac{2}{\pi}\cos^{-1}\left(\frac{l}{2R}\right) \\ &\quad \text{for } 0 \leq l \leq 2R. \end{aligned}$$

Therefore, the distribution of the time taken for one attempt is

$$P(X_v \leq t) = \frac{1}{v}\left[1 - \frac{2}{\pi}\cos^{-1}\left(\frac{vt}{2R}\right)\right] \text{ for } 0 \leq vt \leq 2R. \qquad (15)$$

During each attempt, the node reaches the collection station if its path intersects the area covered by the collection station. If r is the radius of the collection station, then the node's path intersects a collection station if $\theta \leq \sin^{-1}(\frac{r}{R})$ and misses the station otherwise. Since θ is the angle of deviation from the center of the circle in either direction, the probability of a node's path reaching the station in a particular experiment is therefore

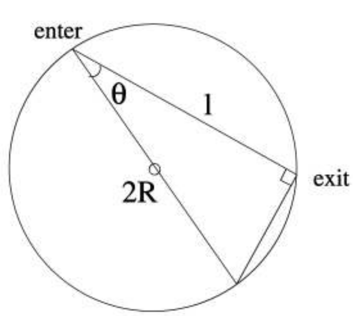

Fig. 9. Node enters circular area at angle θ and travels length l in the range.

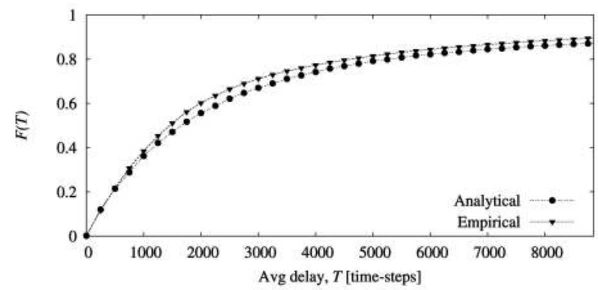

Fig. 10. Distribution of the times until a collection station is reached using fluid mobility.

$$\beta = \frac{2\sin^{-1}\left(\frac{r}{R}\right)}{\pi}.$$

The time taken to reach the station in the first network area "copy" is approximated by $\frac{\mathrm{E}(X_v)}{2}$, where $\mathrm{E}(X_v)$ is the expected value of X_v. We only expect the time to be $\frac{1}{2}$ of $\mathrm{E}(X_v)$ because the node is initially placed uniformly at random in the network area and X_v is the distribution of time across the network area assuming that we begin at the boundary. We also expect that the time spent in the last "copy" of the network (the "copy" where the node actually meets the station) is only $\frac{\mathrm{E}(X_v)}{2}$ because the node travels only half-way across the area. The expected time spent in each other attempt is $\mathrm{E}(X_v)$. Thus, we can write the cumulative distribution for the total time until a station is reached as

$$\begin{aligned} &P\left(T_v \leq \frac{\mathrm{E}(X_v)}{2}\right) = \beta, \\ &P(T_v \leq i\mathrm{E}(X_v)) = (1-\beta)^i \beta \qquad \forall i \in \{1, 2, 3, \ldots\}. \end{aligned} \qquad (16)$$

Up to this point, we assumed that the node moved at speed v. Next, we use (16) to find the cumulative distribution T of the delay for a system whose nodes choose their velocity uniformly at random between $v_{\min}$ and $v_{\max}$. If a node chooses a velocity v' instead of velocity v, then the node's path is the same but is traveled at a different rate. The distribution of times to reach the station is simply a scaled version of T_v, i.e., $P(T_{v'} \leq i) = P(T_v \leq \frac{v'}{v} i)$. Therefore,

$$P(T \leq i) = \frac{1}{v_{\max} - v_{\min}} \int_{v_{\min}}^{v_{\max}} P\left(T_v \leq \frac{v'}{v} i\right) dv'. \qquad (17)$$

In Fig. 10, we show that our analytical calculations agree well with empirical tests for a 300 units by 300 units network area where the node and the station have communication radii of 7 units, $v_{\min} = 0$[units/time-step] and $v_{\max} = 6$[units/time-step]. Note that, using the same methods as above, we are able to find the distribution of the duration of an interaction between a node and a stationary collection station, L_s. This is again the time for a node to cross a circular area along a linear path with velocity chosen from some interval $[v_{\min}, v_{\max}]$. This work can be extrapolated to the duration of an interaction between moving nodes, as shown in [14].

ACKNOWLEDGMENTS

This work was supported in part by the US National Science Foundation (NSF) under grants ANI-0329905 and CNS-0626751, and by the MURI Program administered by the US Air Force Office of Scientific Research (AFOSR) under contract F49620-02-1-0217. The authors thank the reviewers and the editors of the *IEEE Transactions on Mobile Computing* for their assistance in the review process of the paper.

REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks,* vol. 38, pp. 393-442, Mar. 2002.

[2] V.G. Cerf, S.C. Burleigh, A.J. Hooke, L. Torgerson, R.C. Durst, K.L. Scott, K. Fall, and H.S. Weiss, "Delay Tolerant Network Architecture," Internet Draft, work in progress, Jan. 2005.

[3] J. Davis, A. Fagg, and B. Levine, "Wearable Computers as Packet Transport Mechanisms in Highly-Partitioned Ad-Hoc Networks," *Proc. Int'l Symp. Wearable Computing,* pp. 141-148, 2001.

[4] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry, "Epidemic Algorithms for Replicated Database Maintenance," *Proc. Sixth Symp. Principles of Distributed Computing,* pp. 1-12, 1987.

[5] M. Grossglauser and D. Tse, "Mobility Increases the Capacity of Ad-Hoc Wireless Networks," *Proc. INFOCOM,* pp. 477-486, 2001.

[6] Z.J. Haas and T. Small, "A New Networking Model for Biological Applications of Ad Hoc Sensor Networks," *IEEE/ACM Trans. Networking,* vol. 14, no. 1, pp. 27-40, Feb. 2006.

[7] S. Jain, R.C. Shah, W. Brunette, G. Borriello, and S. Roy, "Exploiting Mobility for Energy-Efficient Data Collection in Sensor Networks," *Proc. IEEE Int'l Symp. Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt),* 2004.

[8] Q. Li and D. Rus, "Sending Messages to Mobile Users in Disconnected Ad-Hoc Wireless Networks," *Proc. MobiCom,* pp. 44-55, 1998.

[9] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic Routing in Intermittently Connected Networks," *Proc. ACM SIGMOBILE Mobile Computing and Comm. Rev.,* pp. 19-20, 2003.

[10] B. Liang and Z.J. Haas, "Predictive Distance-Based Mobility Management for PCS Networks," *Proc. INFOCOM,* pp. 1377-1384, 1999.

[11] P. Nain, D. Towsley, B. Liu, and Z. Liu, "Properties of Random Direction Models," Technical Report RR-5284, INRIA, 2004.

[12] C. Perkins, *Ad Hoc Networking,* first ed. Addison-Wesley, 2001.

[13] M.R. Pearlman, J. Deng, B. Liang, and Z.J. Haas, "Elective Participation in Ad Hoc Networks Based on Energy Consumption," *Proc. IEEE Global Telecomm. Conf./Symp. Ad-Hoc Wireless Networks (GLOBECOM/SAWN '02),* pp. 26-31, 2002.

[14] P. Samar and S.B. Wicker, "On the Behavior of Communication Links of a Node in a Multi-Hop Mobile Environment," *Proc. MobiHoc,* pp. 145-156, 2004.

[15] R. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: Modeling a Three-Tier Architecture for Sparse Sensor Networks," *Proc. IEEE Int'l Workshop Sensor Network Protocols and Applications,* 2003.

[16] T. Small and Z.J. Haas, "The Shared Wireless Infostation Model—A New Ad Hoc Networking Paradigm (or Where There Is a Whale, There Is a Way)," *Proc. MobiHoc,* pp. 233-244, 2003.

[17] T. Small and Z.J. Haas, "Designing Markov Models for Mobile Sensor Systems," *Proc. First IEEE Upstate NY Workshop Comm. and Networking,* pp. 172-176, 2004.

[18] T. Small and Z.J. Haas, "Resource and Performance Tradeoffs in Delay-Tolerant Wireless Networks," *Proc. ACM SIGCOMM Workshop Delay Tolerant Networking and Related Topics (WDTN-05),* pp. 260-267, 2005.

[19] M. Stemm and R.H. Katz, "Measuring and Reducing Energy Consumption of Network Interfaces in Hand-Held Devices," *Proc. IEICE Trans. Comm.,* pp. 1125-1131, 1997.

[20] A. Vahdat and D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks," Technical Report CS-2000-06, Duke Univ., 2000.

[21] W. Zhao, M. Ammar, and E. Zegura, "A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks," *Proc. MobiHoc,* pp. 187-198, 2004.

Tara Small received the BSc degree from the University of New Brunswick, Fredericton, Canada, in 2000 and the MS and PhD degrees from Cornell University, Ithaca, New York, in 2004 and 2005, respectively. Her research interests are in the areas of wireless communication and mathematical modeling. Recently, she has been studying resource trade-offs in delay-tolerant wireless sensor networks. She is a member of the IEEE.

Zygmunt J. Haas received the BSc degree in electrical engineering in 1979 and the MSc degree in electrical engineering in 1985. In 1988, he received the PhD degree from Stanford University and subsequently joined AT&T Bell Laboratories in the Network Research Department. There, he pursued research on wireless communications, mobility management, fast protocols, optical networks, and optical switching. From September 1994 until July 1995, Dr. Haas worked for the AT&T Wireless Center of Excellence, where he investigated various aspects of wireless and mobile networking, concentrating on TCP/IP networks. In August 1995, he joined the faculty of the School of Electrical and Computer Engineering at Cornell University, where he is now a professor. Dr. Haas is an author of numerous technical papers and holds 15 patents in the fields of high-speed networking, wireless networks, and optical switching. He has organized several workshops, delivered numerous tutorials at major IEEE and ACM conferences, and serves as an editor for several journals and magazines, including the *IEEE Transactions on Wireless Communications*, *IEEE Communications Magazine*, and *ACM/Kluwer Wireless Networks*. He has been a guest editor of three *IEEE Journal on Selected Areas in Communications* issues and served as a chair of the IEEE Technical Committee on Personal Communications. His interests include mobile and wireless communication and networks, biologically-inspired systems, and performance evaluation of large and complex systems. He is a fellow of the IEEE. For more information, see http://wnl.ece.cornell.edu.