

A New Networking Model for Biological Applications of Ad Hoc Sensor Networks

Zygmunt J. Haas, *Senior Member*, and Tara Small, *Member*

Abstract—In this paper, we introduce the *Shared Wireless Infostation Model (SWIM)*, which extends the *Infostation* model by incorporating information replication, storage, and diffusion into a mobile ad hoc network architecture with intermittent connectivity. *SWIM* is able to reduce the delay of packet delivery at the expense of increased storage at the network nodes. Furthermore, *SWIM* improves the overall *capacity–delay* tradeoff by only moderately increasing the storage requirements. This tradeoff is examined here in the context of a practical application—acquisition of telemetry data from radio-tagged whales. To reduce the storage requirements, without affecting the network delay, we propose and study a number of schemes for deletion of obsolete information from the network nodes. In particular, through the use of Markov chains, we compare the performance of five such storage deletion schemes, which, by increasing the computational complexity of the routing algorithm, mitigate the storage requirements. The results of our study will allow a network designer to implement such a system and to tune its performance in a delay-tolerant environment with intermittent connectivity, as to ensure with some chosen level of confidence that the information is successfully carried through the mobile network and delivered within some time period.

Index Terms—Animal monitoring, biological data acquisition, capacity-delay tradeoff, data dissemination, delay-tolerant networks, epidemic modelling, epidemic routing, flooding, sensor networks, sparse networks, stochastic routing, store-carry-forward, *SWIM*, wireless Infostation model.

I. INTRODUCTION

THE CAPACITY of ad hoc networks has been extensively studied in the technical literature. For example, in fixed networks, Gupta and Kumar [11] proved the rather pessimistic result that as the number of nodes, N , per unit area increases, the throughput of each connection decreases at the rate on the order of $1/\sqrt{N}$. Grossglauer and Tse [10] showed that by introducing node mobility and exploiting *multiuser relaying*, the long-term throughput for source-destination pairs can remain constant as the node density increases. In order to provide this constant throughput, messages propagated in their network travel at most two hops: eventually, either the source node can directly reach the destination, or the source sends the message

to a relay node and the relay node is able to directly reach the destination. Of course, to support a constant throughput, the packet delay may be arbitrarily large. Arpacioglu and Haas [2], [27] established that generally a desired end-to-end throughput is not achievable as N increases unless the average number of hops between a source and a destination is bounded by a constant, and the area grows with N at the rate $O(A^{\min(\gamma/2, 1)})$ if $\gamma \neq 2$, or $O(A/\log A)$ if $\gamma = 2$, where A is the network area and γ is the radio propagation decay exponent.

Grossglauer and Tse designed their constant-throughput system [10] based on the idea that maximizing throughput involves scheduling transmissions over sufficiently good quality paths.¹ Since information is not passed until the nodes are sufficiently close to one another, their scheme can be viewed as constructing a *virtual link* between the source and destination nodes, a virtual link that is enabled by nodes' mobility. The *Infostation* model proposed by researchers at WINLAB at Rutgers University exploits an idea where users are able to connect only within a small distance from ports or *Infostations* [7], [9], [13]. These *Infostations* are distributed throughout the network coverage area, offering only geographically intermittent coverage, but with very high signal quality that allows very high data rates for communication between nodes and *Infostations*.

In this paper, we evaluate a new communication paradigm, which we term the *Shared Wireless Infostation Model (SWIM)*. The *SWIM* model introduces information sharing among the network nodes through the processes of replication, storing, and diffusing of information throughout the network nodes. In fact, the mobile nodes serve as physical carriers of information within the *Infostation* environment. The *SWIM* model uses virtual links created by mobility, similar to the model in [10], as well as physical links between nearby nodes. Thus, it is essentially a marriage of the *Infostation* model with the *ad hoc networking* technology [12]. The nodes function cooperatively in creation of a network, for example, a sensor network [1], [21], [22], in order to improve the delivery of packets to their destinations. When, due to their mobility, two nodes come into communication distance, the nodes exchange their stored information, storing a single copy of each packet at each node. Later, when one of the network nodes carrying the packet reaches the vicinity of a collection station, the packet is offloaded to the collection station. Thus, only one replica of the packet needs to reach only one collection station to be successfully offloaded. Since only one of the replicas of each packet needs to be offloaded, *SWIM* allows the network to reduce the delay at the expense of, what we will see later is, a reasonable increase

Manuscript received February 5, 2004; revised December 29, 2004; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Palazzo. This work was supported in part by the National Science Foundation under Grants ANI-0329905 and ANI-0081357, and the DoD Multidisciplinary University Research Initiative (MURI) program administered by the Air Force Office of Scientific Research under Grant F49620-02-1-0233. This work is based in part on an earlier work presented at the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing, June 2003.

The authors are with the Department of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853 USA (e-mail: haas@ece.cornell.edu; tsmall@cam.cornell.edu).

Digital Object Identifier 10.1109/TNET.2005.863461

¹Meaning, short paths, with path length of at most two hops.

in the storage requirement.² To increase the probability that a packet is recovered from the network, a number of collection stations can be strategically distributed throughout the network coverage area. Distribution of the collection stations should be done in a way that maximizes the chances of packets being offloaded.³

II. SWIM NETWORK OF WHALE TAGS

Motivated by the fact that education about marine mammals, their environment and their preservation is currently a key concern for many groups, such as the “Save the Whales” organization, we chose to illustrate the SWIM framework by applying SWIM to the practical problem of data acquisition from radio-tagged whales [26]. Miniature electronic sensing and transmitting tags are implanted into whales, for the purpose of collecting biological or environmental data. These data need to be recovered from the tags and delivered to the marine biologists, which is usually done using radio telemetry [19].

The use of radio tags for habitat monitoring and animal tracking has existed for many years.⁴ However, recent technological advances allow sufficient miniaturization of the sensors, so that many sensing units can be deployed at any time, creating the opportunity for formation of sensor networks [16]. Sensors can be deployed prior to the animal breeding season, or at another less sensitive period, and data can be retrieved by propagating within the sensor network to data collection points. Furthermore, data retrieving and sensor programming could be done with little human intervention. In this way, the sensor networks are less disruptive to the environment, allowing less biased measurements. It is also more economical, since with sensor networks less personnel is needed to monitor the systems.

In the *Infostation* model, users can connect to collection stations, which provide strong radio signal reception in small and disjoint geographical areas and, as a result, offer very high data rates to users in those areas. However, due to the lack of continuous coverage, these high data rates come at the expense of providing intermittent connectivity only. This may cause significant delay in the delivery of packets since a node that wishes to transmit data may be located outside the coverage area of the stations for an extended period of time. To reduce delay, these systems trade delay for capacity by varying the degree of connectivity and by exploiting the mobility of the nodes.

Although some increased delay can be tolerated in the whale radio tag application, if the delay is too long, the data will likely be lost. The tags, like the one shown in Fig. 1, are foreign objects injected into the whales, which are typically expelled from the host’s body within 3 to 3 1/2 months, and are lost. Moreover, the amount of collected data prevents the storage of data from prolonged measurement periods without data offloading. Therefore, data retrieval must occur through transmissions from

²Note that much of a node’s storage is occupied by other nodes’ information. Thus, such a system requires high degree of cooperation among the network nodes.

³By placing the collection stations near areas that are frequented often by the nodes.

⁴UK Wildlife-Tracking. <http://www.wildlife-tracking.co.uk/>, and H.A.B.I.T. Research. <http://www.habitresearch.com/>

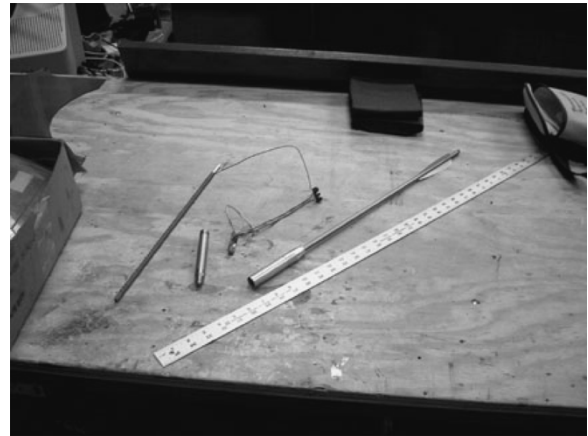


Fig. 1. Whale tag prototype, to be delivered using a crossbow.

the tag while it remains attached to the whale. If we were to use the *Infostation* model, where a user must physically travel to the vicinity of a collection station to communicate, this could lead to a significant delay in our whale tag application. Thus, to address our application requirements and to reduce the information offloading delay, we extend the *Infostation* model by allowing sharing of data among the network nodes. This is the essence of SWIM—through the sharing of information among the network nodes, the nodes are used as carriers to effectively diffuse the information throughout the network, so that the retrieval of the information can be done in a more timely manner.

The radio tags of a SWIM network form a partially-connected network and are designed to collect sensor data continuously and to store these data in time-stamped packets that can be copied between nodes. For example, desirable physiological data concerning the whales include body temperature, heartbeat, and diving depth, while environmental data include illumination level, temperature, and salinity of the water. The radio tags are custom made with Texas Instruments MSP430F149 microprocessors, which allow monitoring of many different sensors, reprogramming in the field, and optimistically have a range of 1–2 km.

SWIM is an example of a delay-tolerant network, where paths rarely exist with all their links being available at the same time. This type of network is recently gaining increased interest, as evident by the Delay Tolerant Networking Research Group which was chartered by the Internet Research Task Force in October 2002. In their Internet draft, Cerf *et al.* discuss the architecture for delay tolerant networks based on the functionality of the U.S. Postal Service. Some systems, such as the Zebanet project [14] and the Manatee project [3], have been studied, which like SWIM explore the use of mobility to facilitate communication. In such systems with sparse networking topologies, small groups of tags are able to communicate, but most of the other tags remain isolated with no connectivity. Occasionally, tags get close to collection stations and offload their data.

Clearly, by allowing the copies of the packet to be stored within the memory of the mobile nodes, the time until one of the replicas reaches a collection station can be significantly reduced. However, this comes at a price; spreading of the packets

to other nodes consumes network resources, such as capacity and storage. Thus, again, we are faced with the *capacity–delay* tradeoff. SWIM lets us control this tradeoff by controlling the parameters of the spread, for example, the probability of packet transmission between two adjacent nodes, the transmission range of each node, or the number and distribution of the collection stations. In this paper, we examine the tradeoff between the amount of storage used at the nodes and the network delay, and we compare it with the traditional *Infostation* model, for a particular reduction in delay.

In Section III, we first develop methods to analytically calculate the probability distribution of the packets’ lifetime in the SWIM system. Section IV confirms that these analytical distributions match well with the actual empirical distributions of the networks. In general, we expect the storage capacity necessary for the SWIM model to increase, relative to the traditional *Infostation* model, since in the SWIM model packets are copied on many nodes, although the time necessary to store a packet, before it is offloaded is much smaller. Therefore, the overall and relative storage requirements of the two schemes cannot be easily predicted. We illustrate the tradeoff between storage and packet delay in Section V.

Section VI shows that on an individual packet level, the storage requirement is further improved by schemes that wisely discard obsolete packets. For example, a node may selectively discard the information in its memory, when there is sufficiently high probability that the information has already been offloaded to one of the collection stations by another node. A node may also retain the identifier of the packet that it offloaded, so that in the future it would refuse information that it had stored previously. Section VII examines the entire system at once, by incorporating individual packet results into queueing methods, to obtain a distribution for the number of packets in a node’s queue. These results allow a network designer to appropriately size the storage of a node, as to account for variation in the queue sizes and to ensure that at most a small fraction of the packets is lost due to buffer overflow.

III. DATA PROPAGATION MODEL

Data that is collected on a whale tag is stored locally. As a whale comes in close proximity to another whale, the stored information is transmitted with some “probability of packet transmission,” p , and is stored in the recipient whale tag’s memory as well. As the whales migrate throughout the system, a whale that surfaces and comes in close contact with one of the collection stations, offloads all the data in its memory⁵ onto the collection station. Appropriately placing collection stations close to feeding areas, allows the devices to upload the data at high data-rate for longer periods of time as the whales feed and socialize near the surface of the water.

If the packets are shared with probability $p = 1$, then whenever two whales are close to each other all packets are transmitted, so we experience the largest delay reduction and the highest increase in storage requirement of the system. Sharing

with probability $p = 0$ represents the original *Infostation* architecture. Thus, by sharing packets with probabilities between 0 and 1, SWIM can achieve different instantiations of the tradeoff between network capacity and network delay. The performance of the system for varying values of p is examined in [23], however, due to space limitations, we will consider in this paper only the case of links with sufficient capacity, i.e., $p = 1$.

Typically, the collection stations are placed on buoys and float on the water.⁶ To meet the delay requirements, several collection stations should be placed within the habitat. After receiving and storing the information from the whales, the collection stations transmit the information to shore, either by coordination with other collection stations,⁷ or directly to a satellite,⁸ whenever a satellite passes overhead.⁹ Collection stations could alternatively be placed on seabirds, flying high above the water. Such collection stations would then be mobile, and the data would be gathered from the mobile collection stations at the known roosting grounds of these seabirds.

The lifetime of a packet in the network varies considerably depending on the mobility patterns of the whales, which are specific to the species of whales under consideration. One might expect nearly daily surfacing near collection stations for whales off the coast of the Hawaiian islands, leading to delays on the order of hours. Whereas, migratory whales may only visit known feeding grounds twice a year, so packets may be in the system on the order of months. Clearly, the SWIM system needs to be designed accordingly to bring down the offloading time to a sufficiently small value.

In order to study the lifetimes of packets, which we define as the time from packet creation until its first offloading, we model the propagation of each packet of data information generated by a whale tag as the spread of one infectious disease. Epidemiological models have been previously studied for networks [18], [24], concentrating on the time from the initial outbreak of a disease until all infected subjects have been healed and on threshold probabilities of outbreaks of epidemic messages. Here, we study metrics that might be more relevant to a network engineer, such as the delay until the first copy of a packet is offloaded, but we are not interested in what happened thereafter.¹⁰ A whale is “infected” (in state I) if it has a copy of the data packet stored in its memory. A whale is “susceptible” to infection (in state S) if it does not yet have a copy stored in its memory, but could potentially acquire the packet from another whale. A whale is “recovered” or healed from the disease (in state R) if it has offloaded the packet to a collection station. A packet can be stored at most once on each tag (one cannot be infected multiple times with

⁶Note that the communication between whale-embedded radio tags should use acoustic transmission, alleviating the need for the whales to surface to communicate.

⁷Forming another network of a “higher tier.”

⁸The buoys consist of two parts, the above-the-water part, which communicate with the satellite through radio frequencies, and the below-the-water part, which communicates with the radio tags through acoustic frequencies.

⁹Note that the use of satellite for collecting data from collection stations rather than from the radio tags is significantly easier, as the collection stations are always above the water and no time synchronization between the transmitter and the receiving satellite is required.

¹⁰This is analogous to the case of a single disease and the time at which one subject recovers from the disease.

⁵Whether its own data or data from other whales.

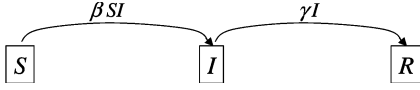


Fig. 2. Markov chain model of an infectious disease with susceptible, infected, and recovered states.

the same disease). By storing the unique identifiers of the previously received packets, a whale may become “immune” to receiving the same packet again. Using this modeling of the packet sharing, we are able to use formulae from epidemiology to find the probability that a packet is offloaded (the node is “healed”) as a function of the time it has spent in the system.

In Fig. 2, the $S(t)$ represents the number of whales in the state S at time t , $I(t)$ represents the number of whales in the state I at time t , and $R(t)$ represents the number of whales in the R state at time t . β is the average contact rate between two whales, per whale. The recovery rate, labeled as γ , is the per-whale rate of contacts between a whale and any collection station. Since the transition into state I has the rate βSI and the transition out of state I has the rate γI ,¹¹ we can write the following differential flow-preservation equation:

$$\frac{dI}{dt} = \beta SI - \gamma I \quad (1)$$

to represent this system. Note that once a copy reaches R (i.e., it has been offloaded), the rates change. So the S-I-R model is invalid after that time. The S-I-R model is investigated further in [5].

We assume that there are M randomly placed collection stations in our network of coverage area of A , and that two whales, or a whale and a collection station, can communicate with one another if the distance between them is not larger than r or r_{cs} , respectively. Let T be a random variable representing the time that a packet spends in the system before it is offloaded, i.e., the time from the packet creation until its first copy is offloaded to a collection station. Since initially we consider only one unique packet in the system, at time $t = 0$ only one whale carries the packet, and since any one of the N whales is either in the state S or in the state I , then

$$\begin{aligned} S(0) &= N - 1, & I(0) &= 1, & S(t) + I(t) &= N, & \text{for } t < T \\ R(t) &= 0, & & & & & \text{for } t < T \text{ and } R(T) = 1 \end{aligned}$$

By solving the differential (1), applied to the rates of the Markov chain in Fig. 2, we find an analytical form for $I(T)$. Then, assuming independent increments, we derive another differential equation for the cumulative distribution function $F(T)$, which represents the probability that the packet is offloaded by time T . We solve the equation using the initial condition that $F(0) = (\pi r^2 M / \text{system area})$, which is the probability of a whale being placed within the range, r , of any of the M collection stations at time $t = 0$. The formula for the resulting function $F(T)$ is given by

$$F(T) = 1 - K \left(\frac{N - 1}{e^{\beta NT} + N - 1} \right)^{\frac{\gamma}{\beta}} \quad (2)$$

¹¹Note that the parameter γ grows linearly with the number of collection stations.

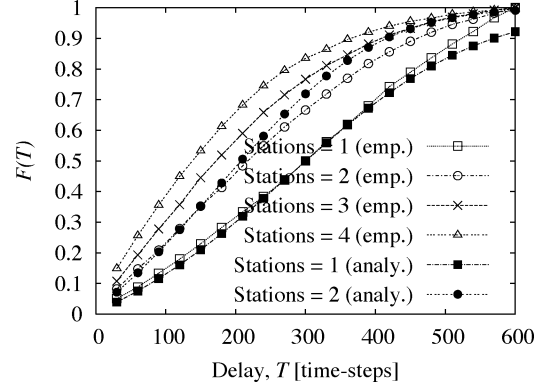


Fig. 3. Probability functions of T , the time from packet creation until offloading, for different numbers of collection stations in the system.

where $K = [(N - 1)/N]^{-\gamma/\beta} [1 - (\pi r^2 M/A)]$.

For example, if $F(300) = 0.5$, this means there is probability 0.5 that a packet is offloaded in 300 time-steps or less. By using the inverse of this function, we can choose a desired probability¹² P_{thresh} and find the value T_p for which $T_p = F^{-1}(P_{\text{thresh}})$. This means that by the time T_p , the packet will be offloaded with probability P_{thresh} . After time T_p , the packets are erased from the system and are no longer considered; therefore the “loss probability” of a unique packet in the system is equal to $1 - F(T_p)$.

IV. SIMULATING THE NETWORK DELAY

We model the network area as a torus, i.e., a rectangular area with edges that wrap around. Thus, a whale that swims off the right edge re-enters at the left edge; similar wrap exists for the top and the bottom edges. The mobility patterns of the whales determines the values of the contact parameters β and γ in (2). We use a simple mobility pattern, the random linear mobility, which is a variant of the *random waypoint* model [4] with pause time 0. This pattern is used to examine some common $F(T)$ properties. In the simulation, the whales swim in straight lines for a fixed number of time-steps, s , with a randomly chosen speed up to v_{max} and in a random direction. Every s time-steps, a new speed and a new direction are chosen for each whale.

At the beginning of the simulation, one whale carries the only replica of the packet. At every iteration, if a whale carrying the packet is within the infection range of another whale, the packet is replicated at the other whale. If any whale carrying the packet is within infection range of a collection station, then the simulation is stopped, and the time, T , is recorded as the lifetime of the packet. The simulation is run many times, and the lifetime statistics is compiled, resulting in the empirical probability function $F(T)$. As one would expect, the $F(T)$ curves are steeper (representing shorter delay) as the number of whales and as the number of collection stations increase. Fig. 3 shows the empirical $F(T)$ curves for different numbers of collection stations, $M = 1, 2, 3, 4$. In this example, swimming speeds of the whales are chosen between 0 to 6 units per time-step on a 300 units by 300 units toroidal area, the reception radius r of the whale tags is 7.5 units, the reception radius of each station r_{cs} is 15 units,

¹²We refer to P_{thresh} as confidence level.

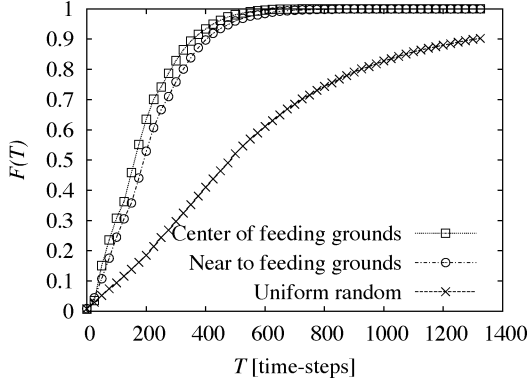


Fig. 4. Effect of different collection station arrangements on the cumulative distribution $F(T)$.

and the number of time-steps in a consistent direction, s , for the random linear mobility model is 15. In order to validate the empirical $F(T)$, we determine the contact rates of β and γ from the simulation. Then we use these contact rates in the theoretical formula of $F(T)$, (1). The analytical curves for the theoretical $F(T)$ are also shown in Fig. 3. Notice that most of the deviations between the empirical and analytical $F(T)$ distributions occur due to the averaging of many whale-whale contact rates into one constant term β . More accurate calculations where β is expressed as a function of the system state will be introduced in Section VI.

A more realistic mobility model captures the physical whale behavior [17] by incorporating feeding grounds. In this enhanced model, three issues govern the direction of the whales' positions at any time: migration in a specified direction, grouping of whales, and direction to the nearest feeding ground. Females tend to group together with other females, while grown males tend to be more solitary in their behavior and group with females, but not with other males [6]. Whales are attracted to feeding grounds, when they are hungry. Inside the feeding grounds, whales move slowly and sometimes stop. When a whale becomes less hungry, it can leave the grounds for a significant time before returning. In our simulation, direction for the whales' mobility is determined by a weighted vector sum of the direction of migration, of the direction to the nearest female, and of the direction the nearest feeding area.

Since the whales are attracted to the centers of the feeding grounds, they are more likely to offload their packet if the collection station is placed inside the feeding grounds. Such placing of the collection stations can significantly reduce the delay. This is demonstrated in Fig. 4 by comparing the "Center of feeding grounds" and the "Near to feeding grounds" curves to the "Uniform random" curve. Obviously, the location of the collection stations is a very significant parameter in the performance of the system. The grouping of whales can also significantly affect the delay, since more grouping promotes more sharing of data and a higher number of copies in the system reduces the time until the first copy is offloaded to a collection station. However, more storage is required.

Up to now, we have assumed that the collection stations are fixed in their locations. Another possible model considers mobile collection stations; for example, collection stations mounted on seabirds that glide above the ocean along the

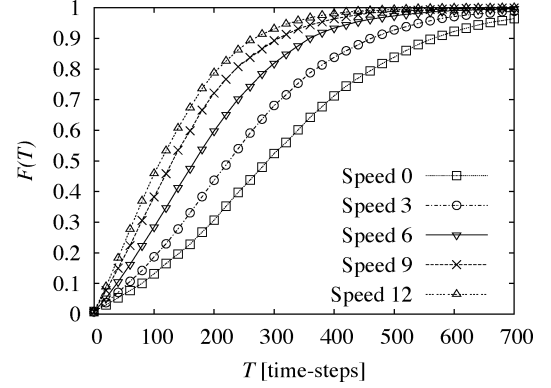


Fig. 5. Cumulative distribution curves with varying speeds of the mobile collection stations.

turbulent air above the waves. Fig. 5 shows that increasing the speed of the mobile collection stations has a positive effect on the packet offload time, when both collection stations and whales use the random linear mobility pattern. As there is more opportunity for contact as the speed of the collection stations increases, the value of T_p is reduced and, therefore, the storage requirements are reduced as well. Note that the trends observed in this set of curves are highly dependent on the relative mobility patterns of the whales and the collection stations.

V. STORAGE REQUIREMENTS

Equipped with the $F(T)$ curves, the information about the contact rate between the whales, and the contact rate between the whales and the collection stations, we are able to calculate the expected storage usage for all the copies of a unique packet for a desired confidence level of the packet delivery, P_{thresh} . As an example, suppose that the designer specifies $P_{\text{thresh}} = 0.9$, then $T_{0.9} = F^{-1}(0.9)$ is the time necessary to wait in order to achieve offloading with probability of 0.9. This should be the "expiration time" of the packet and its replicas, which is the value of the *Time-To-Live (TTL)* field. The *TTL* is inserted in the packet at the time of creation and decreased at every time-step. When the packet is passed to another node, the remaining value of *TTL* is passed on to the replica. Note that this eliminates the need for a global clock, because the *TTL* is adjusted by each node individually. When $TTL = 0$, any replica of the packet remaining in the network is discarded.

A quick, though somewhat naïve, approach for calculating the storage usage for the system involves estimation of the packet lifetime and use of Little's formula to calculate the average number of packets in the system at the offloading time. The number of packets in the system at offloadings is a random variable, in general, with its variance greater than 0. However, we choose to use the average number in the system as a first order approximation estimate of the actual value.¹³

As an example, suppose that 10 whales are tagged and placed uniformly at random at each time-step, in an area of 900 km^2 with 1 collection station. The reception range of the radio tags is 1.4 km and the reception range of the collection stations is 3 km. This can be modeled as a system with $N = 10$ whales, and $M = 1$ collection station. From the corresponding $F(T)$

¹³More precise values for storage usage in the system will be calculated later.

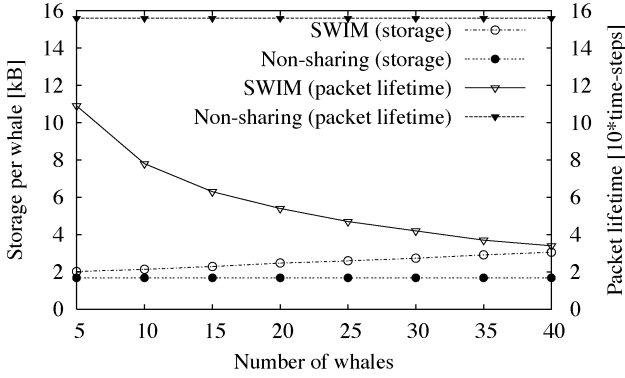


Fig. 6. Necessary storage requirements and times packets remain in the system using SWIM versus the non-sharing model.

curve, we find that $F^{-1}(0.9) \approx 78$. The “expiration time” of the packets should therefore be set to 78 time-steps. Suppose further that each whale generates a packet every 30 time-steps. Using Little’s formula with generation rate $\lambda = 1/30$ packets per time-steps per whale, the expected number of all the packet replicas in the system is $E(P) = (\# \text{ of whales})\lambda T_{0.9} = 26$ [packets].

An estimate of the number of copies of each packet in the system, $E(I)$, is the expected number of whales infected with the packet at the time of offloading. It can be shown from our simulation that $E(I) = 2.55$ in this case. This number assists us in the calculation of the storage used by every radio device, which is (duplicates)(different packets)(bytes/packet) = $E(I) * E(P) * (330 \text{ bytes/packet}) = 2.14 \text{ kB/whale}$.

Recall that in this example, the probability of sharing packets between close-by whales is 1, so the results correspond to the largest delay decrease and the largest storage usage of the SWIM model. Fig. 6 shows that the increase in storage is very reasonable for the achieved, relatively large, decrease in delay. The advantage of SWIM is even more pronounced as the number of whales increases.

In Fig. 6, the pure *Infostation* model with no data replication is termed the “non-sharing case.” While in this case, the per-whale storage usage remains the same as the number of whales increases, the storage usage in the SWIM case grows slightly due to more packet replications. However, this increased storage is somewhat mitigated by the fact that the duration of the packet’s storage in the system is reduced as well, i.e., in SWIM, there are more packet copies in the network, but they remain for a shorter time. Thus, the overall resulting increase in the storage is mild.

The packet lifetime for the non-sharing system is constant over the different numbers of whales, since more whales in the system offer no advantage, i.e., every whale must itself reach the collection station to offload its packets. However, SWIM replicates packets among the network nodes, so for more whales, there will be more copies of a packet in the network. Thus, SWIM achieves smaller delays as the number of whales increases.

In practice, one would want to include additional storage in the memory calculation to account for statistical variability in the number of packets stored in a tag.¹⁴ This additional storage, not accounted for in our simple approach in this section, will

be considered when we reexamine our storage evaluation in the following sections.

VI. PACKET DISCARDING STRATEGIES

There are numerous methods that could be used to model the packet generation, storage, and discarding. We consider here five possible methods: JUST_TTL, FULL_ERASE, IMMUNE, IMMUNE_TX, and VACCINE. These methods are progressively more complex, while at the same time progressively improving the performance of one another. In all of these methods, the original packet and all of its copies that exist in the system are discarded $T_p = F^{-1}(P_{\text{thresh}})$ time-steps after the original packet was created. The JUST_TTL method has no other features than the discarding at time T_p of all the copies, which remain in the system by that time. The FULL_ERASE method discards the copy of the packet completely from the offloading node just after it has been offloaded to a collection station. In this scheme, a whale is able to receive the packet, offload and discard it, then receive that packet again from another whale. Like FULL_ERASE, the IMMUNE method discards the packet when it is offloaded, but keeps an identifier of the offloaded packet, so it will not accept that packet again. We refer to this identifier as an “antipacket,”¹⁵ since it prevents re-infection with the packet. IMMUNE_TX discards the packet when offloaded, keeping the antipacket, like IMMUNE, but it also shares this antipacket with other whales that carry copies of that packet. This means that if a whale is infected with the packet, it may receive an identifier “antipacket” from a second whale saying a copy of the packet has already been offloaded. The packet would then be discarded, and the identifier kept on the first whale. Finally, the VACCINE method shares all of the packets and the antipackets when within transmission range, whether the packet is already stored at the receiving whale or not.

The simple model of Fig. 2 is not adequate to represent these complicated schemes and possibly elaborate whale mobility patterns, so we choose to extend it to Markov chains that have more precise representation of the contact parameters and the system state. From this point on, we will use the parameters Δt_i , for $1 \leq i \leq N - 1$ to capture the dependence of the value of the rate parameters on the system state.¹⁶ Δt_i is the average time interval between the state of the system with i infected whales and the state of the system with $i + 1$ infected whales, without the packet being offloaded. To determine the values of Δt_i for our mobility and system models, we ran a simulation of the mobility pattern numerous times and recorded the time from when the system entered the state $I = i$ until it enters the state $I = i + 1$. The average of these times gave us the Δt_i . Next, we formulate discrete Markov chain models representing the discarding methods described earlier. It is assumed in these models that at most one replication event occurs per time-step of the simulation. Let Δt equal the time of one time-step in the simulation, which we make small enough, so that the condition $\Delta t_i \gg \Delta t$ holds. This ensures that only one event occurs per one time-step. If $\Delta t_i \approx \Delta t$, or $\Delta t_i < \Delta t$, then our models

¹⁵Similar to *antibody* of a biological agent.

¹⁶Rather than the single contact rate, β .

¹⁴Since the loss due to buffer overflow is not negligible.

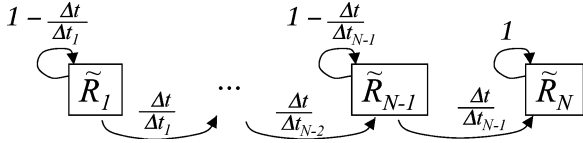


Fig. 7. JUST_TTL Markov chain: no invalidation.

become invalid, since transitions between nonadjacent states need to be included. In what follows, we discuss the states and the transitions among the states for the five sharing/discarding methods of SWIM.

A. Method 1: JUST_TTL

As shown in Fig. 7, the states in this model are all labeled as \tilde{R}_i , $1 \leq i \leq N$, where \tilde{R} indicates that no packet is marked as “offloaded to a collection station” or “recovered.” Even if a copy of the packet is offloaded, the packet is not marked and the replicas continue to exist in the network. The variable i indicates the number of copies of the packet in the system, e.g., a system in state \tilde{R}_3 would have 3 copies, meaning the original packet would have been shared twice.

The mean time from state \tilde{R}_i to \tilde{R}_{i+1} is Δt_i , so this means that the transitions from state \tilde{R}_i to state \tilde{R}_{i+1} occur with probability $\Delta t / \Delta t_i$ in the discrete Markov chain. We can calculate the probabilities of state \tilde{R}_i at time k , $P(\tilde{R}_i, k)$ for all $1 \leq i \leq N$, and $k > 1$. The initial conditions are $P(\tilde{R}_1, 0) = 1$ and $P(\tilde{R}_i, 0) = 0$ for all $i \neq 1$. The model is valid for T_p time-steps, after which, all copies of the packets are discarded from the system, so all of the probabilities are then 0.

$$P(\tilde{R}_i, k) = \begin{cases} P(\tilde{R}_1, k-1) \left(1 - \frac{\Delta t}{\Delta t_1}\right), & \text{for } i = 1 \\ P(\tilde{R}_{i-1}, k-1) \frac{\Delta t}{\Delta t_{i-1}} \\ \quad + P(\tilde{R}_i, k-1) \left(1 - \frac{\Delta t}{\Delta t_i}\right) \\ P(\tilde{R}_{N-1}, k-1) \frac{\Delta t}{\Delta t_{N-1}} \\ \quad + P(\tilde{R}_N, k-1), & \text{for } i = N \end{cases}$$

B. Method 2: FULL_ERASE

We realize that when a packet copy has been offloaded once, future offloadings are unproductive. For this reason, packets may be discarded as soon as they are offloaded to a collection station. In Fig. 8, states \tilde{R}_i , $1 \leq i \leq N$ are needed as in JUST_TTL, however, now new states R_i , $0 \leq i \leq N$ are also necessary. The R_i states represent i packets remaining in the system and at least one copy of the packet having been offloaded, while the \tilde{R}_i states represent no packets having been offloaded to a collection station. For example, a system in the state \tilde{R}_4 could offload one of its copies and enter the state R_3 . Next, the remaining copies might be shared with other whales, entering the states R_4 and then R_5 . Once the system has left any of the tilde states, \tilde{R}_i , it cannot re-enter to a tilde state again.

For simplicity in notation, let $g = \gamma \Delta t$, and let $p_i = (\Delta t / \Delta t_i)(1 - ig)$ for this and for the following methods. We notice that the transition probability from the state \tilde{R}_i to the state \tilde{R}_{i+1} is $p_i = (\Delta t / \Delta t_i)(1 - ig)$, since this is the conditional probability that the packet has been

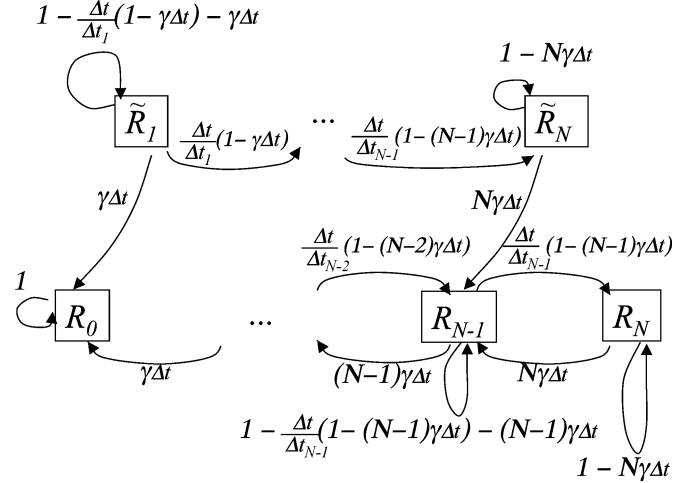


Fig. 8. FULL_ERASE Markov chain: collection stations mark packets to discard.

shared, but not offloaded at that time-step. First, consider the probabilities of the states $\tilde{R}_1, \dots, \tilde{R}_N$, i.e., the states where no collection station has yet received any copy of the packet. These probabilities are the same for all of the FULL_ERASE, the IMMUNE, the IMMUNE_TX, and the VACCINE cases.

$$P(\tilde{R}_i, k) = \begin{cases} P(\tilde{R}_1, k-1)(1 - p_1 - g), & \text{for } i = 1 \\ P(\tilde{R}_{i-1}, k-1)p_{i-1} \\ \quad + P(\tilde{R}_i, k-1)(1 - p_i - ig) \\ P(\tilde{R}_{N-1}, k-1)p_{N-1} \\ \quad + P(\tilde{R}_N, k-1)(1 - Ng), & \text{for } i = N \end{cases}$$

Next, consider the states R_i , which represent i packets in the system and at least one collection station having received a copy of the packet.

$$P(R_i, k) = \begin{cases} P(\tilde{R}_1, k-1)g + P(R_0, k-1) \\ \quad + P(R_1, k-1)g, & \text{for } i = 0 \\ P(\tilde{R}_2, k-1)2g \\ \quad + P(R_1, k-1)(1 - p_1 - g) \\ \quad + P(R_2, k-1)2g, & \text{for } i = 1 \\ P(\tilde{R}_{i+1}, k-1)(i+1)g \\ \quad + P(R_{i-1}, k-1)p_{i-1} \\ \quad + P(R_i, k-1)(1 - p_i - ig) \\ \quad + P(R_{i+1}, k-1)(i+1)g \\ P(\tilde{R}_{N-1}, k-1)p_{N-1} \\ \quad + P(R_N, k-1)(1 - Ng), & \text{for } i = N \end{cases}$$

C. Method 3: IMMUNE

In addition to the \tilde{R}_j states of the previous cases, where the collection station had not yet received any copy of the packet, we need each R state’s label to express not only the number of packets, but also the number of identifiers (including full packet and antipacket identifiers) present in the system. In these states R_j^i , i indicates the number of entire packets in the system and j indicates the number of identifiers (packets + antipackets). This means that when a system in state \tilde{R}_j offloads a packet, it enters state R_j^{j-1} because the packet is discarded from the node when the packet is offloaded, but its antipacket identifier is created and

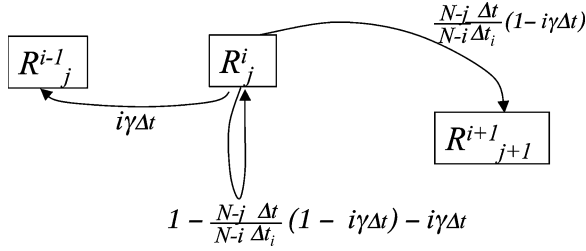


Fig. 9. IMMUNE Markov chain: collection stations mark packets for deletion, but headers remain as antipackets.

retained in the system. Because of the complexity of the Markov chain in this case, we show only a representative state in Fig. 9.

The \tilde{R}_j states and their transition probabilities behave as in the FULL_ERASE case, and after entering an R_j^i state, the system can never return to a tilde state, \tilde{R}_j , again. The three possible events and their corresponding transition probabilities, when the system is in the R_j^i state, are as follows.

- **New whale is infected**

In the previous method, the probability of transition from state R_i to R_{i+1} represents a new whale infection in a system with i whales infected and $N - i$ whales susceptible. In this method, antipackets are introduced, so a transition R_j^i to R_{j+1}^{i+1} begins with i whales infected, but only $N - j$ susceptible. Since only $N - j$ are susceptible, instead of $N - i$, the transition probability for this method is reduced compared to the previous method by the factor of $(N - j)/(N - i)$.

- **Infected whale is recovered**

One of the i whales which still have the entire packet may pass by a collection station with probability ig and have the packet discarded, keeping only the identifier, so the system enters state R_j^{i-1} .

- **No event**

With the remaining probability $1 - ((N - j)/(N - i))p_i - ig$, the system stays in state R_j^i .

We now write down the transition probabilities of the R_j^i states. First, we realize that only states R_j^i with $i < j$ can have nonzero probability. There cannot be more identifiers (packets + antipackets) than packets in the system, and if the system has entered an R_j^i state, then at least one of the packets must have been offloaded.

For $j = 1$,

$$P(R_1^0, k) = P(\tilde{R}_1, k - 1)g + P(R_1^0, k - 1)$$

and for other fixed $j < N$,

$$P(R_j^i, k) = \begin{cases} P(R_j^1, k - 1)g + P(R_j^0, k - 1), & \text{for } i = 0 \\ \begin{aligned} &P(R_j^2, k - 1)2g + P(R_j^1, k - 1) \\ &\times \left(1 - \frac{(N - j)}{(N - 1)}p_1 - g\right), & \text{for } i = 1 \\ &P(R_{j-1}^{i-1}, k - 1)\frac{(N - j + 1)}{(N - i + 1)}p_{i-1} \\ &+ P(R_j^{i+1}, k - 1)(i + 1)g \\ &+ P(R_j^i, k - 1)\left[1 - \frac{(N - j)}{(N - i)}p_i - ig\right]. \end{aligned} \end{cases}$$

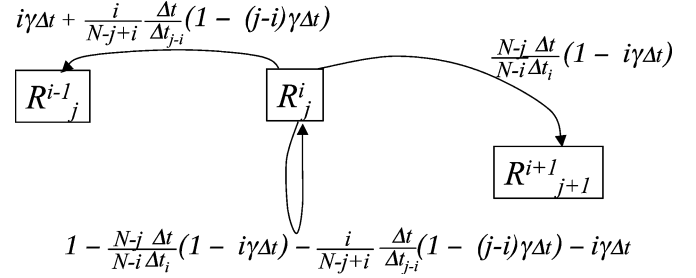


Fig. 10. IMMUNE_TX Markov chain: antipacket headers remain and are back-propagated to previously infected whales.

For $j = N$, all $i < j - 1$ are the same as above, except for

$$P(R_N^{N-1}, k) = P(R_{N-1}^{N-2}, k - 1)\frac{p_{N-2}}{2} + P(R_N^{N-1}, k - 1)[1 - (N - 1)g].$$

Finally, we have not considered transitions from the states \tilde{R}_j to a state R_j^i . In particular, for every $i = j - 1$, we need to add the extra term $P(\tilde{R}_j, k - 1)jg$.

D. Method 4: IMMUNE_TX

This scheme is very similar to the previous case and is represented by the same states, as seen in Fig. 10, but uses slightly different transition probabilities. Again, the \tilde{R}_i states behave as above and after entering an R_j^i state, the system can never return to a tilde state, \tilde{R}_i , again. The new infection events are the same as in the IMMUNE case; only the recovery event is different.

- **Infected whale is recovered**

One of the i whales which still have the content of the packet may pass by a collection station with probability ig or could also pass by one of the $j - i$ whales that carry the packet. The spreading of the copies of an antipacket is similar to the spreading of a packet. $j - i$ antipackets exist and can be spread, so $N - (j - i)$ whales do not carry the antipacket. We would expect that the probability of spreading would be p_{j-i} , however only i of the $N - (j - i)$ whales are able to receive the antipacket. Therefore, the probability is reduced by the factor of $i/(N - (j - i))$, so $i/(N - j + i)$ is the probability the system enters the state R_j^{i-1} .

- **No event**

With the remaining probability $1 - ((N - j)/(N - i))p_i - (i/(N - j + i))p_{j-i} - ig$, the system stays in state R_j^i .

The \tilde{R}_i states have the same probabilities as in the case of the FULL_ERASE and the IMMUNE methods. The R_j^i states have the same probabilities as in the IMMUNE case, but there are required terms that address the back propagation of antipackets by the whales. For all $i, 1 \leq i < j$,

$$P(R_j^i, k) = P(R_j^i, k)_{\text{IMMUNE}} + \mathbb{1}(i < j - 1) \times P(R_j^{i+1}, k - 1)\frac{(i + 1)}{N - j + (i + 1)}p_{j-(i+1)} - \mathbb{1}(i > 1)P(R_j^i, k - 1)\frac{i}{N - j + i}p_{j-i}$$

where $\mathbb{1}(x) = 1$ if x is true, and $\mathbb{1}(x) = 0$ otherwise.

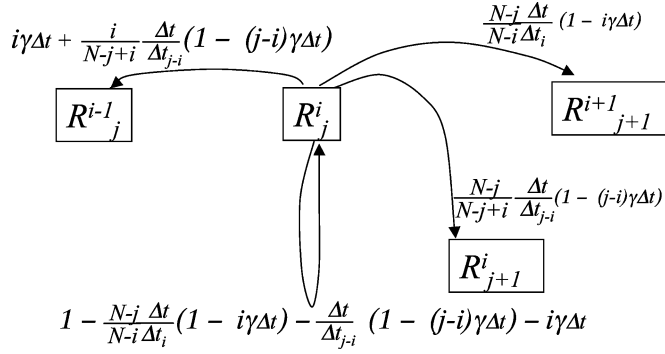


Fig. 11. VACCINE Markov chain: antipacket headers back-propagated to any whales.

E. Method 5: VACCINE

In the previous cases, whales could only carry the immunizing antipacket if they had carried the entire packet previously. In this case, a whale that has not been previously infected with the packet will become immune (store just the antipacket identifier) by receiving the antipacket from an immunized whale, without ever being infected with the actual packet. This event is represented by the transition probability $((N - j)/(N - (j - i)))p_{j-i}$ from the state R_j^i to the state R_{j+1}^i in Fig. 11, since $j - i$ antipackets exist, and only $N - j$ are susceptible out of the $N - (j - i)$ other whales. Receiving an antipacket without ever having stored the actual packet represents a transition to the state R_j^{i-1} .

Therefore, the \tilde{R}_i states have the same probabilities as in the cases of the FULL_ERASE, the IMMUNE, and the IMMUNE_TX methods, and for all i ($1 \leq i < j \leq N$) the extra terms are as follows:

$$\begin{aligned}
 P(R_j^i, k) &= P(R_j^i, k)_{\text{IMMUNE}} \\
 &+ \mathbf{1}(i < j-1) P(R_j^{i+1}, k-1) \frac{(i+1)}{N-j+(i+1)} p_{j-(i+1)} \\
 &+ \mathbf{1}(i > 0) P(R_{j-1}^i, k-1) \frac{N-(j-1)}{N-(j-1)+i} p_{(j-1)-i} \\
 &- \mathbf{1}(N-j > 0) P(R_j^i, k-1) \frac{N-j}{N-j+i} p_{j-i} \\
 &- \mathbf{1}(i > 1) P(R_j^i, k-1) \frac{i}{N-j+i} p_{j-i}
 \end{aligned}$$

where $\mathbf{1}(x) = 1$ if x is true, and $\mathbf{1}(x) = 0$ otherwise.

F. Average Number of Copies

We now proceed with the evaluation of the storage usage in the system in the above five cases. In each of these cases, we calculate the time-average of expected storage values over the states. For example, in the three latter cases, the average number of packets in the system is

$$\overline{P(T_p)} = \left(\frac{1}{T_p} \right) \sum_{t=1}^{T_p} \left[\sum_{i=1}^N \left[iP(\tilde{R}_i, t) + \sum_{m=1}^N iP(R_m^i, t) \right] \right].$$

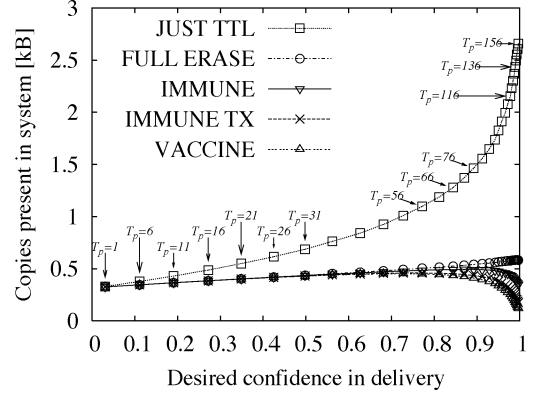


Fig. 12. Expected storage required for 10 whales, varying probability of delivery and indicating corresponding expiration times.

Recall that in some methods, there may be antipackets as well as packet headers, so we must calculate the average number of headers (packets + antipackets) separately:

$$\overline{H(T_p)} = \left(\frac{1}{T_p} \right) \sum_{t=1}^{T_p} \left[\sum_{i=1}^N \left[iP(\tilde{R}_i, t) + \sum_{m=1}^N mP(R_m^i, t) \right] \right].$$

These values are affected by the mobility and network models, which are represented through the $N - 1$ parameters Δt_i . In Fig. 12, $\Delta t_1 = 36.37$, $\Delta t_2 = 20.32$, $\Delta t_3 = 15.77$, $\Delta t_4 = 13.85$, $\Delta t_5 = 13.49$, $\Delta t_6 = 13.83$, $\Delta t_7 = 16.01$, $\Delta t_8 = 20.53$, and $\Delta t_9 = 35.70$. Compiling the data for each model, we calculated the average amount of storage for the replicas of the packet and its antipacket identifiers for a given time. Call this average storage value $EI(T_p)$, where T_p is calculated for a desired probability of offloading, $F(T_p)$:

$$E(I(T_p)) = \frac{326}{330} \overline{P(T_p)} + \frac{4}{330} \overline{H(T_p)} \text{ [bytes]}$$

since packets in this application are 330 bytes long, consisting of a 4-byte header and 326-byte packet contents. Identifiers are assumed to be of the size of the packet header. Shown in Fig. 12 is the parametrical plot of the average storage usage as a function of the desired confidence (offloading probability), $(F(T_p), EI(T_p))$. This figure also shows selected values of the parameter, T_p , in the JUST_TTL case. Notice that for methods IMMUNE, IMMUNE_TX, and VACCINE the average storage begins to decrease at high confidence level. This rather surprising effect is due to the dependence of the confidence level on T_p . As the confidence level approaches 1, the necessary time T_p for the packet to remain in the system increases; eventually $T_p \rightarrow \infty$ as $F(T_p) \rightarrow 1$, increasing monotonically. In the methods IMMUNE, IMMUNE_TX, and VACCINE the packet identifiers prohibit the whales from storing a copy of the packet again, so eventually as T gets large, nearly all the whales refuse storage of the packet, and the average required storage is thereby reduced.

Using these five methods, we can also depict the storage-delay tradeoff using SWIM. We fix the desired $P_{\text{thresh}} = 0.9$, then to reduce the delay, we increase the sharing of the packets by increasing the density of whale tags in the system. Fig. 13 exhibits the storage-delay tradeoff due to this increased sharing;

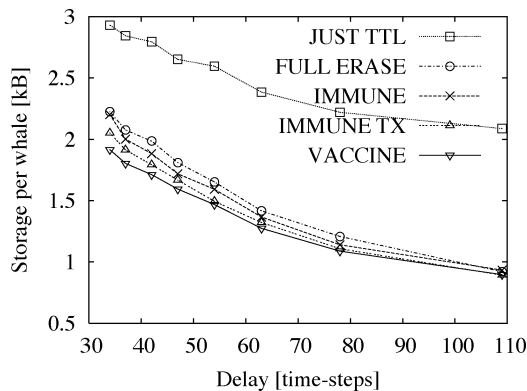


Fig. 13. Storage–delay tradeoff of SWIM using the different methods of discarding packets.

clearly, to achieve shorter delay, one must invest more storage in the system.

Let us briefly discuss the complexities of these models. The processing complexities of these algorithms are relatively small, so we only consider the communication complexity. We define this communication complexity as the number of bits that need to be transmitted in the system so that the protocol is successfully carried out. For all methods except FULL_ERASE, the process is $O(N)$, that is, the whales will receive a copy of an individual packet at most once, and (in the latter cases) copy of an antipacket at most once. The situation is different for FULL_ERASE, because a whale tag can offload its packet copy and erase it, then receive a copy again. Therefore, the only limiting factor on the number of transmissions in the FULL_ERASE method is the TTL , so the method is $O(TTL)$.

Let us consider the average efficiency of the different packet discarding strategies in the SWIM system and define the concept of an optimally-delivered packet. We say that a packet copy is optimally-delivered, at a specific time, if it is the first packet copy that is delivered to a collection station at that time. We will call this packet *optTime*. In a similar manner, we can define *optPower* as the packet copy that is offloaded within the TTL , which has traveled the fewest number of hops (and therefore its path used the least power in the system). We can also define *optStor* as the packet copy that is offloaded within the TTL , but used the least storage if we consider only the copies that followed that particular path to the destination.

We wish to compare the global resource consumption of SWIM to the resource consumption of ideal algorithms which would result in these optimal packets. Clearly, it is impossible to know which packet copy will be optimal until the simulation is actually completed, so in practice, we cannot achieve the actual resource consumption of these optimal packets. However, conceptually, the optimal packets provide a lower bound for the minimum resource consumption that we need in the system to achieve desired performance. Furthermore, the optimal packets can serve as benchmarks to normalize actual performance metrics.

Many metrics could be used to compare the global resource usage to the optimal case, but we choose to use power. Let the global power metric be the sum of the number of times the packet is copied between any two nodes and the number of

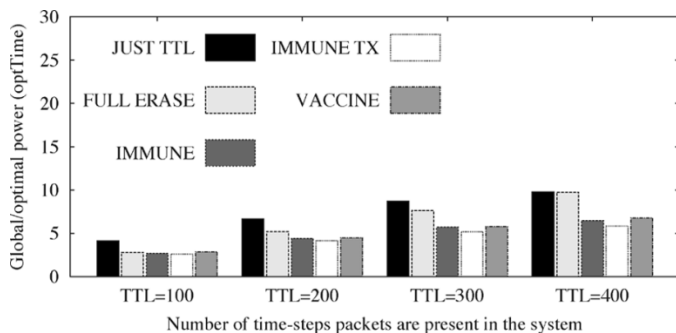


Fig. 14. Power metric of resource consumption ratio of the global compared to the *optTime* optimal packet.

nodes that transmit a packet copy to a collection station. The optimal packet metric is defined similarly, but we only consider the power used for the packets along the path followed by the optimal copy and assume that there is no power used after the optimal copy is offloaded. Therefore, the power metric for the optimal node is the number of hops, h , on the optimal path ($h-1$ transmissions between nodes and *one* transmission to the station).

For each of the methods of discarding packets, Fig. 14 graphs the value of the fraction (global metric/optimal metric) using four different values for the TTL of the packets. We see similar trends in the graphs for all types of optimal packets, so we discuss the *optTime* graphs only. The system is less efficient in terms of power for all methods of packet discarding if the TTL of the packets is larger. This is a very intuitive result. When we allow more time for the packets to remain in the system (increasing TTL), this allows for more opportunities to copy the packet and to use resources after one of the copies is already offloaded. Note that this graph shows the ratio of the global metric to the optimal metric given that a packet is offloaded, so the graph shows that there is more resource consumption for larger TTL , but is not representing the benefit of the larger TTL value.

For a particular TTL , we can examine the relative power usage of the different offloading strategies. We see that in the strategies that use immunity, (IMMUNE, IMMUNE_TX, VACCINE,) the power consumed is less than JUST_TTL and FULL_ERASE. This is simply due to the fact that the antipackets reduce the number of packet copies in the system for the packets that have already been offloaded. IMMUNE_TX is more efficient than IMMUNE, since IMMUNE_TX is more effective in reducing the number of copies of offloaded packet. VACCINE reduces the number of offloaded packet copies even more than IMMUNE_TX, but it does not have larger efficiency because it spends more energy on transmitting antipackets to all nodes.

VII. REPRESENTING THE ENTIRE SYSTEM

The previous sections describe methods for sharing and discarding the copies of a single packet. Here, we study properties of the entire system, where each whale tag generates unique packets periodically, then passes and/or discards each of them following the previous schemes.

Equipped with the mean storage requirement for each individual packet, $E(I(T_p))$, it is reasonably easy to compute the mean storage requirement per whale (in kB) for the entire system, $E(I(T_p)) * \lambda * T_p * ((330/1024) \text{ kB/packet})$, since each packet is assumed to have the size of 330 bytes or 330/1024 kB. This does not, however, provide an indication of the variance of the number of packets stored on each whale tag. If an attempt is made to send packets, but the receiver buffer is full, then all such new packets are dropped. We can account for this dropping probability as a reduction in p , the probability of successful offloading in the single packet model, but we would like this reduction to be small, so that there is not much additional loss of packets.¹⁷ This is achieved by increasing the buffer size, so the probability of the buffer overflowing is very small. In particular, we calculate the expected mean storage, μ , then include additional storage to safeguard against overflow, where the additional storage is relative to the variance of the buffer occupancy, σ^2 . For example, we could choose the size of the buffers to be $\mu + \sigma$ or $\mu + 2\sigma$. The example shown later in Fig. 16 chooses buffer sizes as $\mu + \sigma$ and leads to a probability of packet dropping of 15%. If we increase the additional storage amount and make the buffers larger at the nodes, then the delay is reduced, because there are more copies of each packet in the system (since fewer copies need to be overwritten due to the finite buffer size). However, now the storage increases and so does the energy to transmit more packets.

To be able to determine σ^2 , we need the probability distribution of the number of packets in each whale tag i , Q_i , in the infinite buffer case. To assist us in the solution of Q_i , we model the system as an imaginary global queue that, at each point in time, contains all the packets present in the system. In particular, let $Q = \sum_{i=1}^N Q_i$ represent the number of all the copies of all the different packets in the system, i.e., the number of packets in the global queue. However, due to the complex nature of the global queue, we employ an approximation: we assume that the arrival of all the copies of a packet to the global queue occurs at the time of the original packet creation, rather than at times when the packet is actually replicated from one whale to another. This is a conservative approximation for the purpose of evaluation of the variance of Q_i , since in reality the arrival of the copies of a packet will be spread in time, reducing the variance. We further assume that the number of replicas of the packet to arrive to the global queue is the maximum number of the packet copies present in the system for a particular run. For the JUST_TTL case, packets are replicated when they are shared between whales, but are not removed from the system until the expiration time. Thus, at time $T_p = F^{-1}(P_{\text{thresh}})$, the number of copies of a unique packet in the system is a maximum. Using other methods, the maximum number of packets may occur at values smaller than T_p .

A. Simulating the Global Queue

The steady state of the global queue can be calculated empirically through simulation, where the simulation generates packets periodically for every whale in the system, given the

¹⁷If p is reduced, there are fewer transmissions overall, so the energy is reduced, but the delay increases.

set of periods and their offsets in time. Let $I(t)$ be a random variable representing the distribution of the number of packets in the system at time t . Let $t = t_{\text{max}}$ be the time at which the expected value, $E(I(t))$, is maximal. In the simulation, random numbers are sampled from the distribution $I(t_{\text{max}})$ to provide the size of the batch arrivals, described below Section VII-B. $I(t_{\text{max}})$ is a random variable, representing the number of copies in the system when the expected number of copies is maximal. The batch of packet copies sampled from the $I(t_{\text{max}})$ distribution is added to the global queue, Q , as soon as the original packet is generated and all existing copies are removed at time T_p . We see that by using a stochastic value sampled from $I(t_{\text{max}})$, we are conservatively estimating the number of copies and modeling a situation where all of these copies exist in the system for the entire time $[0, T_p]$. Clearly, the number of copies is $0 \leq I(t_{\text{max}}) \leq N$, since at most all the whales can be infected with the packet. After the simulation ends, the sample mean and sample variance of the number of packets in the global queue at steady-state are calculated.

B. On the Theory of the Global Queue

Alternatively, we can also calculate the theoretical solution of the global queue, with the same batch arrivals of size sampled from the $I(t_{\text{max}})$ distribution. The arrivals remain in the system for the entire time $[0, T_p]$. When the number of whale tags is moderately large and the arrival processes of new packets at different tags have slightly different periods, the arrivals of groups of packets act like a Poisson queue with batch arrivals. The system is said to have infinitely many servers, since all the packets are “served” at the same time,¹⁸ meaning that all packets enter the system immediately and can be shared or offloaded at any time-step. A Poisson queue with batch arrivals involves groups of customers which reach servers with i.i.d. exponentially distributed interarrival times. The numbers of customers in these groups is determined by the distribution function $I(t_{\text{max}})$. The service times in this case are deterministic (i.e., constant time T_p). Finally, since there are infinitely many servers available, customers never have to wait in the global queue, i.e., the only delay is due to the deterministic service time.

Let us first consider a Markov chain for the arrivals only and incorporate the service times afterward. The chain of arrivals represents a pure birth process with “batch births” and with states: 0, 1, 2, 3, ... These states represent the total number of customers (packets) in the system. The arrival of a new set of customers is represented by a transition to a higher numbered state. These are transitions between state i and $i + x$, where x is a discrete random variable with distribution $I(t_{\text{max}})$, and represents the birth (arrival) of a batch of size x .

C. Chain With Batch Arrivals

Let $P_k(t)$ represent the probability that k customers are in the system at time t . Assume that the largest possible batch size is N and let λ be the rate of arrival of the batches. The probability of a batch containing i customers is represented by g_i , where $g_i = 0$ for $i > N$, and the Markov chain representing the system is

¹⁸Service time is the time during which a packet resides on a tag. Since we assume that all packets are created at the time of the original packet generation, all service times are equal to T_p .

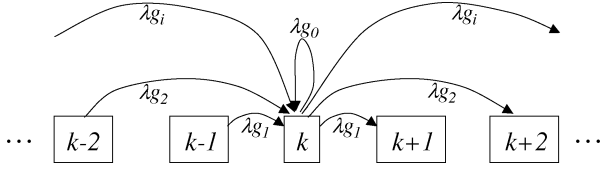


Fig. 15. Poisson batch arrival process.

shown in Fig. 15. We write the following differential equations representing the rate of change of the states' probabilities:

$$\frac{dP_0(t)}{dt} = -\lambda(1-g_0)P_0(t) \quad (3)$$

and for all $k \geq 1$,

$$\frac{dP_k(t)}{dt} = -\lambda(1-g_0)P_k(t) + \sum_{i=\max(0, k-N)}^{k-1} P_i(t)\lambda g_{k-i} \quad (4)$$

where we assume that $P_k(0) = \begin{cases} 1 & k=0 \\ 0 & \text{otherwise.} \end{cases}$

From (4), we are able to solve for $P_0(t)$ immediately, i.e., $P_0(t) = e^{-\lambda(1-g_0)t}$. This $P_0(t)$ is then substituted into (4) and using an integrating factor, we solve for $P_1(t)$. Iteratively, we calculate $P_2(t), P_3(t), \dots$. Below, we list the solution of the first four of these probabilities:

$$P_0(t) = e^{-\lambda(1-g_0)t}$$

$$P_1(t) = e^{-\lambda(1-g_0)t}(\lambda g_1 t)$$

$$P_2(t) = e^{-\lambda(1-g_0)t} \left[\frac{(\lambda g_1)^2}{2!} t^2 + \lambda g_2 t \right]$$

$$P_3(t) = e^{-\lambda(1-g_0)t} \left[\frac{(\lambda g_1)^3}{3!} t^3 + (\lambda g_1)(\lambda g_2) t^2 + (\lambda g_3) t \right].$$

We see an obvious pattern in $P_k(t)$ related to combinatorics. Consider, for example, $P_4(t)$. It has five terms, each related to different partitions of the number 4 into its additive terms: $4 = 1 + 1 + 1 + 1 = 1 + 1 + 2 = 1 + 3 = 2 + 2$. The first partition has four 1's. Each of those 1's corresponds to the g_1 probability of a 1-step Poisson transition, and we can say that g_1 has the frequency of 4 in this term. This corresponds to the Poisson factor $((\lambda g_1)^4/4!)t^4 e^{-\lambda g_1 t}$ in the solution for $P_4(t)$. Since it has no 2-step transitions, there is the factor $((\lambda g_2)^0/0!)t^0 e^{-\lambda g_2 t} = e^{-\lambda g_2 t}$ in this term as well. Similarly, absence of 3-step or 4-step transitions give $e^{-\lambda g_3 t}$ and $e^{-\lambda g_4 t}$ factors in this term. This means that the resulting term is $((\lambda g_1)^4/4!)t^4 e^{-\lambda t}$. The next partition is $1 + 1 + 2$, so g_1 has frequency 2, g_2 has frequency 1, and g_3 and g_4 each have frequency 0. This results in the corresponding term

$$\frac{(\lambda g_1)^2}{2} (\lambda g_2) t^3 e^{-\lambda t} = \frac{(\lambda g_1)^2}{2!} t^2 e^{-\lambda g_1 t} * \frac{(\lambda g_2)^1}{1!} t^1 e^{-\lambda g_2 t} * \frac{(\lambda g_3)^0}{0!} t^0 e^{-\lambda g_3 t} * \frac{(\lambda g_4)^0}{0!} t^0 e^{-\lambda g_4 t}.$$

Methods of efficiently finding these partitions exist, but we omit further discussion on this topic due to space limitation. By relating each of these partitions to a term in the equation for the probability $P_k(t)$ using the method described above, we were able to relatively simply determine the $P_k(t)$ values, which represent the sought probabilities of k customers arriving into the queue by time t .

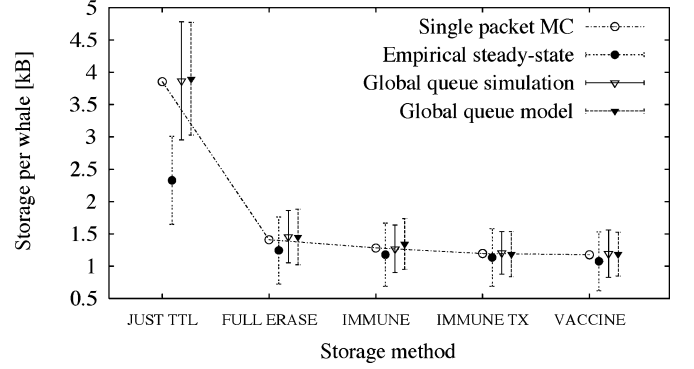


Fig. 16. Mean storage present per whale with error bars indicating one standard deviation for a 10-whale system.

D. Poisson Batch Arrivals With Service

Let $P_k^*(t)$ represent the probability that k customers are being served at time t in a system with Poisson batch arrivals and deterministic service time T_p . Equipped with the probabilities $P_k(t)$ for the Poisson batch arrival process, finding these $P_k^*(t)$, for all $k \geq 0$, is particularly easy.

If $t \leq T_p$, then all batch arrivals still remain in the system at time t , because their service time would not have elapsed. Thus, $P_k^*(t) = P_k(t)$. Alternatively, if $t > T_p$, then any arrivals before time $t - T_p$ would already have been serviced, so we consider only customer arrivals in $(t - T_p, t)$. Since this process is stationary, this value is the same as the number of customers arriving in $(0, T_p)$. Therefore,

$$P_k^*(t) = \begin{cases} P_k(t), & \text{if } t \leq T_p \\ P_k(T_p), & \text{if } t > T_p. \end{cases}$$

Clearly, we consider $P_k^*(t) = P_k(T_p)$ to be the steady state probabilities of the numbers of packets in the system. The above is, indeed, a solution of the batch-arrival M/D/ ∞ queue.

E. Packet Distributions in the Tags

By assuming that all of the buffers are i.i.d. with respect to the number of packets they carry, we can simply divide the batch-arrival M/D/ ∞ queue solution by the number of whales to find the distribution of the number of packets on each individual tag. This provides not only the mean number of packets on a tag, which is already known from the single packet Markov chain, but any quantile that the designer wishes to use in order to ensure that the buffers at nodes could handle some level of variance without overflowing.

Fig. 16 compares the numbers of packets in the 10-whale system for the four different evaluation methods described earlier. The random mobility pattern is assumed. The first method (Single Packet MC) uses the individual packet Markov chain to find $E(I(t_{\max}))$ and then estimates the total storage usage in kilobytes through $N * E(I(t_{\max})) * \lambda * T_p * ((330/1024) \text{ kB/packet})$, described earlier, to give a somewhat conservative estimate of the mean number of packets in the system, μ . All of the other curves measure storage by considering the entire system at once, choosing the total buffer sizes to be $\mu + \sigma$. This leads to a probability of packet blocking of 15%.

The second metric (Empirical Steady-State) is the average number of packets on a tag measured in simulation of multiple packets in steady state. This is an empirical measurement of the actual number of packets in the system, rather than the conservative estimate used in the other methods. For this reason, the curve of the second metric is lower than the other curves. The third metric (Global Queue Simulation) uses the simulation of the global queue with batch arrivals with distribution $I(t_{\max})$. Finally, the fourth metric (Global Queue Model) calculates the probabilities analytically using the batch Poisson method with deterministic service times. All of these values are divided by the number of whales in the system to provide a value of storage per whale. The latter three methods also provide confidence intervals of one standard deviation, since these methods supply the entire distribution of the number of packets stored in the system. By comparing these models, we conclude that the Global Queue Model gives a sufficiently good approximation to reality and thus could be used to estimate the queue variance, without using the more sophisticated mathematical model and without resorting to simulations.

By adding one standard deviation of the storage distribution on a whale tag to its mean, and since the global queue estimate is a conservative one,¹⁹ we can ensure that the packet loss in the queues on the nodes is not too high, so that it does not contribute significantly to the overall loss of packets *vis-à-vis* the offloading confidence level. Using this estimate, the storage usage per whale for JUST_TTL is 4.77 kB/whale, for FULL_ERASE is 1.89 kB/whale, for IMMUNE is 1.73 kB/whale, for IMMUNE_TX is 1.54 kB/whale and for VACCINE is 1.53 kB/whale. Of course, if a higher degree of protection is necessary, than one could add more than one standard deviation of the storage distribution in the calculation of the size of the buffer.

We conclude that even when accounting for the variability in the tag's queue, the storage requirements remain quite reasonable for a practical implementation.

VIII. CONCLUDING REMARKS

In this paper, we have presented and evaluated a new communication paradigm, the *Shared Wireless Infostation Model*. As opposed to established links, as in a cellular network, SWIM creates what we referred to as virtual links in a network. SWIM exploits mobility in the system, producing those virtual links to offload data when a node is close to a collection station. The virtual links are realized by physical links when nodes come into close contact, to cooperatively propagate information to the destinations through replication of the information on the network nodes.

This communication paradigm has a broad range of applications, such as in the area of telemetry collection and sensor networks. It could be used for animal tracking systems, for medical applications with small sensors to propagate information from one part of the body to another or to an external machine, and to relay traffic or accident information to the public through the

vehicles themselves (e.g., for AMBER alert to relay information regarding missing children to electronic billboards in an area), as well as many other applications. We have chosen to exemplify SWIM using the problem of biological data acquisition from a whale tracking system.

We first analyzed lifetimes of packets in the whale system by equating the propagation of one packet to the propagation of a single infectious disease. Using the theoretical distribution of packet lifetimes provided here, a researcher should be able to choose parameters of the system, so that a large fraction of the data is retrieved before the tags are rejected and lost. Furthermore, we examined the storage usage on each tag in the sharing system and developed mathematical models for five different methods of discarding obsolete packets. Comparing a system of five whale tags to one with 40, we observe that sharing reduces the delay by a factor of 4.6 compared to the non-sharing case. Using the simplest technique to discard packets, this reduction in delay increases the mean storage requirement space by 75%. However, by discarding packets wisely using the more complicated VACCINE method, this reduction in delay requires a mean storage increase of only 14% compared to the non-sharing case.

Finally, the variance of the occupancy of the tag queues is important for the buffer sizing, so that only a small fraction of the packets is lost due to statistical variation in the queues. We found a conservative estimate not only of the mean storage requirement on a whale tag, but of the entire distribution of the packets at any time t , including the steady state. We have shown that by using SWIM to share and offload packets and to discard packet copies wisely, it is possible in many cases to significantly reduce delay of data acquisition with only a moderate increase in storage requirements, even when accounting for variations in node's queue size.

REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–442, Mar. 2002.
- [2] O. Arpacioğlu and Z. J. Haas, "On the scalability and capacity of single-user-detection based wireless networks with isotropic antennas," *IEEE Trans. Wireless Commun.*, to be published.
- [3] A. Beaufour, M. Lepold, and P. Bonnet, "Smart-tag based data dissemination," presented at the 1st ACM Workshop on Wireless Sensor Network and Applications, Atlanta, GA, Oct. 2002.
- [4] C. Bettstetter, "Smooth is better than sharp: a random mobility model for simulation of wireless networks," presented at the 4th ACM Int. Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), Rome, Italy, Jul. 2001.
- [5] F. Brauer and C. Castillo-Chávez, *Mathematical Models in Population Biology and Epidemiology*. New York: Springer-Verlag, 2001.
- [6] J. D. Darling, C. Nicklin, K. Norris, H. Whitehead, and B. Würsig, *Whales, Dolphins, and Porpoises*. Washington, DC: National Geographic Soc., 1995.
- [7] R. Frenkiel, B. R. Badrinath, J. Borràs, and R. Yates, "The Infostations challenge: balancing cost and ubiquity in delivering wireless data," *IEEE Pers. Commun.*, vol. 7, no. 2, pp. 66–71, Apr. 1999.
- [8] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "An empirical study of epidemic algorithms in large scale multihop wireless networks," UCLA Computer Science Tech. Rep., UCLA/CSD-TR 02-0013.
- [9] D. J. Goodman, J. Borràs, N. B. Mandayam, and R. D. Yates, "INFO-STATIONS: a new system for data and messaging services," in *Proc. IEEE VTC'97*, vol. 2, 1997, pp. 969–973.
- [10] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 477–486, Aug. 2002.

¹⁹The global queue estimate is conservative, since it assumes addition of all the copies at the packet origination time and estimates the number of copies based on the maximum number of copies in the system.

- [11] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.
- [12] Z. J. Haas, J. Deng, P. Papadimitratos, and S. Sajama, "Wireless ad hoc networks," in *Encyclopedia of Telecommunications*, J. Proakis, Ed. New York: Wiley, 2002.
- [13] A. Iacono and C. Rose, "Infostations: new perspectives on wireless data networks," Rutgers Univ., New Brunswick, NJ, WINLAB Tech. Doc., 2000.
- [14] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet," presented at the 10th Int. Conf. AS-PLOS, San Jose, CA, Oct. 2002.
- [15] L. Kleinrock, *Queueing Systems Volume I: Theory*. New York: Wiley, 1975.
- [16] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," presented at the WSN'02, Atlanta, GA, Sep. 2002.
- [17] B. R. Mate, R. Gisiner, and J. Mobley, "Local and migratory movements of Hawaiian humpback whales tracked by satellite telemetry," *Canadian J. Zoology*, vol. 76, pp. 863–868, 1998.
- [18] C. Moore and M. E. J. Newman, "Epidemics and percolation in small-world networks," *Phys. Rev. E*, vol. 61, pp. 5678–5682, 2000.
- [19] I. G. Priede, "Wildlife telemetry: an introduction," in *Wildlife Telemetry: Remote Monitoring and Tracking of Animals*, I. G. Priede and S. M. Swift, Eds. New York: Ellis Horwood, 1992.
- [20] S. Resnick, *Adventures in Stochastic Processes*. Cambridge, MA: Birkhäuser, 1992.
- [21] 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)," presented at the ACM MOBIHOC'03, Annapolis, MD, Jun. 2003.
- [22] T. Small, Z. J. Haas, A. Purgue, and K. Fristup, "A sensor network for biological data acquisition," in *Sensor Networks*, M. Ilyas, Ed. Cleveland, OH: CRC Press, 2004.
- [23] T. Small, "Modeling tradeoffs in intermittent-connectivity networks," Ph.D. dissertation, Cornell Univ., Ithaca, NY, Aug. 2005.
- [24] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke Univ., Durham, NC, Tech. Rep. CS-200006.
- [25] J. H. van Lint and R. M. Wilson, *A Course in Combinatorics*. Cambridge, U.K.: Cambridge Univ. Press, 2001.
- [26] H. Wang, J. Elson, L. Girod, D. Estrin, and K. Yao, "Target classification and localization in habitat monitoring," presented at the IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP 2003), Hong Kong, Apr. 2003.
- [27] O. Arpacioglu and Z. Haas, "On the scalability and capacity of planar wireless networks with omnidirectional antennas," *Wireless Commun. Mobile Comput., Special Issue on Scalability Issues in Wireless Networks*, vol. 4, no. 3, pp. 263–279, May 2004, to be published.



Zygmunt J. Haas (S'84–M'88–SM'90) received the B.Sc., M.Sc., and Ph.D. degrees, all in electrical engineering, from Stanford University, Stanford, CA, in 1979, 1985, and 1988, respectively.

From 1988 to 1995, he worked for AT&T Bell Laboratories in the Network Research Department. In 1995, he joined the faculty of the School of Electrical and Computer Engineering at Cornell University, Ithaca, NY, where he is now a Professor. He is an author of over 50 refereed journal publications and over 100 conference papers, and holds 15 patents in the fields of high-speed networking, wireless networks, and optical switching. His interests include mobile and wireless communication and networks, performance evaluation of large and complex systems, and biologically inspired systems.

Dr. Haas has organized several workshops, delivered numerous tutorials at major IEEE and ACM conferences, and serves as an editor of several journals and magazines, including the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the *IEEE Communications Magazine*, and the *Kluwer Wireless Networks Journal*. He has served as a Chair of the IEEE Technical Committee on Personal Communications.



Tara Small (S'05–M'06) received the B.Sc. degree from the University of New Brunswick, Fredericton, NB, Canada, in 2000, and the M.S. degree from Cornell University, Ithaca, NY, in 2004. She is currently pursuing the Ph.D. degree in the field of applied mathematics at Cornell University.

Her research interests are in the areas of wireless communication and mathematical modeling. Recently, she has been studying resource tradeoffs in delay-tolerant wireless sensor networks.