Energy-Efficient Routing in Sparse Networks

by Seung Keun Yoon

This thesis/dissertation document has been electronically approved by the following individuals:

Haas,Zygmunt J (Chairperson)

Shmoys,David B (Minor Member)

Wicker,Stephen B. (Minor Member)

ENERGY-EFFICIENT ROUTING IN SPARSE NETWORKS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

In Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Seung Keun Yoon

August 2010

ENERGY-EFFICIENT ROUTING IN SPARSE NETWORKS

Seung Keun Yoon, Ph. D.

Cornell University 2010

Routing in sparse networks results in relatively long packet delivery delays and is unreliable. Thus, sparse networks are not an appropriate networking solutions for applications in which delay delivery is of critical importance. However, some applications can tolerate substantial amount of delay. Networks which can support communication for such applications are referred to as the Delay-Tolerant Networks (DTNs).

Epidemic Routing Protocol (ERP) has been proposed and widely used for routing in DTNs. ERP is based on the concept of packet flooding – a node that carries a packet transmits (i.e., replicates) the packet to every node that it encounters. Thus, the numbers of copies of the packet rapidly increases. This type of routing results in the shortest possible packet delivery delay and in high packet delivery probability. However, these results come with strict assumptions that the network nodes have sufficiently large amount of energy and storage.

In case where the network sequentially routes multiple packets, increasing the number of copies of the packet has its disadvantages. First is the excessive amount of energy used for transmissions. When nodes have limited amount of battery energy, increased number of copies eventually results in faster depletion of the batteries of the network nodes, reduction in the number of active nodes, and in reduced network lifetime. Second is the amount of memory used for packet storage. A node is likely to receive many different packets which will pile up the memory before the node

encounters the sink. When its memory is full, the node should either remove a packet from its memory, or not receive another packet. Either way, this results in less packet copies and decrease of packet delivery probability.

This study analyzes the ERP for two different cases and shows how to overcome the disadvantages by modifying the ERP. In case when the nodes have limited battery energy, minimizing and equalizing the energy expenditure at the nodes efficiently increases the network lifetime. In case when the nodes have limited storage, using linear combinations of multiple packets increase the packet delivery probability without any major cost of energy or storage.

Seung Keun Yoon was born on April 17, 1978 in Korea. He lived with his parents and his sister in Seoul, but he also had chances to live abroad. During his father's sabbatical, who is a professor of Economics, Seung Keun had a chance to live in Chicago in 1988 and Tokyo Japan in 1997. From 1994 to 1996, he went to Seoul Science High School which led him to his first step as an engineer and in 1997 he entered School of Electrical Engineering at Seoul National University.

During his undergrad, he got interested in robotics and sensors while he was visiting University of Tokyo and Waseda University in Japan. He also did internship for SKW Associates Inc. in Santa Clara, CA, as a programmer. Right before graduating, he finished his military duty as a KATUSA (Korean Augmentation Troop to United States Army) in 8th US Army.

After he came to Cornell University, he became much interested in Biological Sensor Networks, Routing Protocols, and Network Coding. He joined the Wireless Networks Lab (WNL) and worked with Professor Zygmunt Haas who showed him how Mathematics could be used to solve many practical problems. During his life in Cornell University as a Ph.D. student, he was also a member of the board of director of Korean Graduate Student Association at Cornell University (KGSAC).

Seung Keun Yoon's research interests are Energy Conservation, Coding Based Routing Protocols, and Body-area Sensor Networks. The main topic of his research is extending network lifetime and improving packet delivery in sparse networks through energy efficient routing, which can be applied to tracking in wilderness and natural disaster surveillance.

To my family

ACKNOWLEDGMENTS

First of all, I would like to say thank you to my advisor Prof. Zygmunt Haas with my deepest appreciation. His supervision during our meetings had so much to learn and was very motivating. Especially, I loved the writing in his office which talks about a god carrying a man during his misery. Everything he has done to me helped me continue in graduate school.

Also many thanks to our Wireless Networks Laboratory research group. Learning and discussing other people's works were very useful and even exciting. I really enjoyed all the conversations I had with the members in our group, especially with Edward Hua, sometimes related to our research and sometimes just our interests. I would also like to thank my committee members, Prof. David Shmoys and Prof. Stephen Wicker for their advice.

Special thanks to my mentors, Kyusoon Lee, Tae Eung Sung, and Seung Youl Lee who helped me in many ways. I wouldn't have made it to the end without them. And many thanks to all my Catan member friends. Thanks to them I really enjoyed my life in Ithaca. Also thanks to my best friends Junwon Jung and Sangjoon Kim who were always there for me when I needed them.

I would like to mention my parents and my sister for their caring and support. I want to thank them for everything they have done to me throughout my whole life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1.   Background

In a wireless mobile sensor network, nodes consume battery energy to transmit data, sense, or move in the network area. With limited amount of energy in the network nodes, the network lifetime is determined by energy consumption. Thus, one way to extend the network lifetime is to reduce the amount of energy used for communication, for example, by reducing the transmission range of the nodes. When the transmission range decreases, so does the number of neighbors that a nodes is able to communicate with at any time. And when the number of neighbors is small enough, connections between mobile nodes in the network become intermittent. A network where the average number of neighbors is less than one is defined as a *Sparse Network*.

There are other reasons for a network to become a sparse network, besides the intentional reduction in the transmission range to preserve energy. For instance, the reduction in transmission range can be dictated by the need for Low Probability of Detection (LPD) and/or Low Probability of Interception (LPI), as is often the case in military communication. Or, due to security restrictions, certain node can communicate only with a small subset of the network nodes. But whatever the reason is for a network to become sparse, routing in intermittently connected networks is facilitated by the mobility of the nodes and by temporal formation of links between nodes that come to proximity of each other. Thus, although the end-to-end path between the source and the destination does not exist in the network at any time, a *virtual* end-to-end path is created by these temporary links and a packet is forwarded from one node to another until it reaches the destination.

Of course, relying on the mobility of the nodes does not guarantee timely delivery of the communications and is appropriate only in special situations where the applications that use the network are not-delay sensitive. This type of networks is often referred to as Delay-Tolerant Networks (DTNs). Examples of potential uses of DTNs are wildlife monitoring [1,2], underwater surveillance [3], inter-planetary communication [4], or remote village messaging [5].

Since nodes in DTN rely on their mobility to relay data packets, the contingency of encountering other nodes becomes an important factor in the performance of a routing protocol. If the behavior of mobile nodes is determined or predictable, future transmissions between nodes can be scheduled ahead of time. However, typically, the behavior of mobile nodes is random and unpredictable. For such cases, a stochastic routing protocol is required, in which each relaying node chooses its next recipient based on some algorithm.

The *Epidemic Routing Protocol (ERP)* [6] has been suggested as a routing protocol for DTN protocol. In ERP, a packet-carrying node replicates the packet on every encountered node. Thus, ERP relies on packet flooding, resembling the model of epidemiology [7-11]. For a given number of active nodes and when the nodes' mobility pattern is totally random, by generating the maximal number of packet replicas, ERP exhibits the shortest possible end-to-end delay. Under such conditions, ERP also maximizes the probability of packet delivery at any particular time. Nevertheless, ERP accomplishes this by excessive, and quite often wasteful, use of the network resources,[1] such as energy and network capacity. The net result is that, in the long run, ERP will deplete the network resources too fast. Depletion of nodes' batteries reduces the total number of active nodes, reduces the packet delivery

---

[1] Unlike in some other works, such as the SWIM protocol [2,15] for example, since there is no feedback in ERP, ERP continues to excessively use the network resources even after the packet has been delivered to the destination.

probability, and, consequently results in premature inability to deliver the minimum required performance to the users' applications.

Several studies have proposed efficient ways to overcome this drawback of ERP [2,12-27]. For example, *SWIM* [2,15] uses a small sized anti-packet to restrict the packet replication. In the *Spray and Wait* routing protocol [17], the total amount of transmission during ERP is restricted by the source node. Controlling packet flow can also restrict packet replication in ERP [6, 19-22]. The reference [6] uses a limited hop algorithm to restrict the set of nodes that is allowed to propagate to peer nodes. The gossip based algorithm, which was proposed in [20], also controls packet flow by transmitting packets to only a fraction of encountered nodes based on a certain probability. Coding-based protocols have been applied as well to wireless networks in various ways. Network coding increases the network throughput [23-31] and it can also improve packet delivery probability [32]. Erasure-coding [33] uses smaller sized packet "fragments" each carrying only partial information of the original packet.

Most of the above works are focused on efficient routing of a single packet or multiple packets at the same time, with nodes carrying sufficiently large amount of battery energy and storage. Due to the unlimited packet replication in ERP, every time a packet is generated, excessive amount of node energy is used for packet transmission and node memory is used for packet storage. Practically, nodes have limited amount of battery energy and storage which will lead to deterioration in the ability of effectively routing packets, particularly when the network sequentially routes multiple packets.

When nodes have limited amount of battery energy, a node will become inactive when its battery energy gets depleted. Unfortunately, the batteries don't get depleted at the same time, since the energy used for transmission is concentrated in only a partial of the nodes. Even if ERP starts with high packet delivery probability, as the number of inactive nodes increases, the packet delivery probability gradually decreases to zero.

When nodes have limited amount of storage, a node will become partially active when its memory is full. When a "fully-stored" node encounters another node, it should either remove a packet from its memory in order to receive another packet, or refuse to receive any packet. Either way, this decreases the packet replication and eventually reduces the packet deliver probability, even when the nodes have sufficient amount of residual battery energy.

Since the energy efficiency of packet routing depends on the packet delivery probability and the energy consumption, it is obviously inefficient to consume excessive energy if the ERP cannot deliver the packet to the sink. The goal of this paper is to maximize the energy efficiency by maintaining a certain threshold of the packet delivery probability as long as possible.

1.2. Dissertation Outline

In Chapter 2, an analytical model for the Epidemic Routing Protocol (ERP) is proposed based on the statistical properties of mobile nodes. This model allows understanding of the packet replication process of ERP. The model also serves as a basis for derivation of the models for the modified versions of ERP – Restricted Epidemic Routing (RER) protocols.

In Chapter 3, several variations of the ERP are proposed that restrict the ERP in the number of generated replications. The fundamental tradeoffs are examined among the three performance parameters: the energy (as expressed by the number of packet copies), the packet delivery delay, and the packet delivery probability. Comparison among the schemes is carried out as to the efficiency in implementation of the tradeoff function between the number of copies (i.e., energy expenditure) and the packet delivery delay for fixed packet delivery probability.

Chapter 4 extends the discussion of Chapter 3 to sequential routing of multiple packets. The threshold of packet delivery probability is defined that allows determination of the *Network Lifetime*. The restriction schemes from Chapter 3 are evaluated in the context of extending the network lifetime, and the most energy efficient scheme is the one with the longest network lifetime.

While Chapter 4 focuses on network nodes with limited battery energy, Chapter 5 focuses on network nodes with limited memory, and extends the basic model of the sequential ERP from Chapter 2, 3, and 4. This model allows understanding of the relation between the converging limit of packet delivery probability and the packet generation rate.

For high packet generation rate, the packet delivery probability decreases below the threshold, making the network inefficient and useless. In Chapter 6, a scheme that applies *Linear Network Coding* to the ERP (LNC-ERP) is proposed. Analysis of the LNC-ERP facilitates better understanding of the advantages and the disadvantages of the scheme, demonstrating the improvement with respect to the basic ERP in terms of packet delivery probability.

Chapter 7 outlines previous related works in comparison to this research. Summary, conclusion and possible future directions of this research are discussed in Chapter 8.

CHAPTER 2

Properties of Epidemic Routing

2.1.   Fundamental Model for Epidemic Routing

In Epidemic Routing from the time when the source node creates a packet, the number of nodes carrying a replicated packet increases. Consequently, the probability that the sink node[2] will encounter a node that carries a packet increases as well. With no restriction on packet replication, eventually all nodes in the network system will carry a replicated packet and the packet delivery probability converges to 1.

The process of Epidemic Routing is similar to the epidemiological process of a virus spreading [7-9]. During epidemic spreading, the number of nodes carrying a packet (the "infected nodes") increases; i.e., every time an infected node encounters an uninfected node ("susceptible node"), the number of infected nodes increases by 1. If there were an infinite number of nodes in the network, the rate of infection would increase linearly with the number of infected nodes. In a network with a finite number of nodes, as the number of infected nodes increases, the number of susceptible nodes decreases,. So the infection rate reaches its maximum value when the number of infected nodes and the susceptible nodes become (approximately) equal. From that time, the number of infected nodes continues to increase, albeit at an infection rate that decreases with time. Eventually, the number of infected nodes stops increasing when all of the nodes become infected.

Assume that the encounter rate of any two particular nodes in a mobile sensor network is Poisson with parameter λ [7]. If there are N nodes in the network, then

---

[2] In a sensor network, there are usually a few (typically one) special nodes, termed *sink nodes,* to which all of packets are to be delivered. If multiple sink nodes are present, it suffices that a copy of a packet is received by one sink only. Sink nodes do not generate or transmit data packets, but only receives packets from the other network nodes.

when the number of infected nodes is *k* and the number of susceptible nodes is *N-k*, the encounter rate between the infected nodes and the susceptible nodes (infection rate η_k) becomes $\eta_k = k(N-k)\lambda$.

As the number of infected nodes, *k*, increases, the infection rate increases as well: *(N-1)λ, 2(N-2)λ, … , k(N-k)λ, …* until *k* reaches *N/2* (or *(N-1)/2*, if *k* is odd). From there on, the infection rate decreases symmetrically as *k* continues to increase, from *(N/2+1)(N/2-1)λ, (N/2+2)(N/2-2)λ, …, (N-1)λ,...*

2.2. Ordinary Differential Equation Model

A simple epidemic model was studied to approximate the number of people infected by a virus in some population of susceptible people. This simple epidemic model is the logistic model of population growth, attributed to Verhulst (1838).

Suppose the total population is *N*, with *x(t)* and *N-x(t)* are the number of infective and of susceptive people at time t, respectively. Assuming that individuals in the population are homogeneously and randomly mixed, we can write

$$\frac{dx}{dt} = \beta x(N-x) \tag{2.1}$$

where *β* is the pair-wise rate of infection. If a susceptive person becomes infected with probability 1 by encountering an infective person, then *β* is the same as the encounter rate λ. This is analogous to the encounter rate of $x(N-x)\lambda$ in Epidemic Routing when there are *x* nodes propagating a packet. Solving the differential equation (2.1) – the so-called logistic growth equation – we obtain that

$$\lambda dt = \frac{dx}{x(N-x)} = \left(\frac{1}{x} + \frac{1}{N-x}\right)\frac{dx}{N}. \tag{2.2}$$

Integrating Equation (2.2) on (0,t) we can solve for *x(t)*,

$$N\lambda \cdot t = \log\frac{x}{N-x} - \log\frac{x_0}{N-x_0}, \qquad x(t) = \frac{x_0 N}{x_0 + (N-x_0)e^{-N\lambda t}}, \tag{2.3}$$

7

where x_0 is the number of the infective people at $t=0$. Starting from the one infective person, i.e., $x_0 = 1$,

$$x(t) = \frac{N}{1 + (N-1)e^{-N\lambda t}}.$$

(2.4)

As we can see, $x(t) \to N$ as $t \to \infty$.

This Ordinary Differential Equation (ODE) model has been thought to be adequate to characterize the spread of an epidemic among a large population. However, even for a large population, this model is not completely accurate; one reason being that x should take only integer values, while in Equation (2.1) the value of x can be real number. Even if we interpret x as a stochastic average, Equation (2.2) still does not hold. For example, suppose that at some time t, $x = n$ with probability p and $x = n+1$ with probability $(1-p)$. Then, at time t the average x is $(n+1-p)$. But according to Equation (2.2),

$$\frac{dx}{dt} = \lambda(n+1-p)(N-n-1+p) = \lambda\{N(n+1-p) - (n+1)^2 + 2p(n+1) - p^2\}.$$

But the average increasing rate of x at time t should be,

$$\frac{dx}{dt} = \lambda\{pn(N-n) + (1-p)(n+1)(N-n-1)\} = \lambda\{N(n+1-p) - (n+1)^2 + p(2n+1)\}.$$

As these two results are different, x, which assumes real values, cannot be interpreted as the average number of infected individuals. To accurately model the system, we need a different method to derive the number of the infective nodes in an Epidemic Routing network.

2.3. Transition Markov Chain Model

In order to estimate the number of packet replicas in the network at a certain time during the progress of epidemic routing, we analyze the packet flooding mechanism. The basic assumption is that the number of encounters between two particular network

nodes follows the Poisson process, where the time interval between encounters has exponential distribution with rate λ. More information on this assumption is provided in Appendix A.
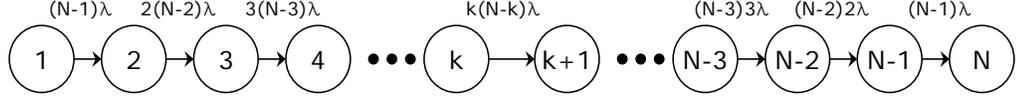
Figure 2.1: The transition diagram of Markov Chain model of packet's copies in the system

Using the encounter rate between infected nodes and susceptible nodes, we derive a transition Markov Chain model for the number of copies in the system, as shown in Figure 2.1. Next, using this model, we derive the probability $P_k(t)$, of k copies of the packet among the N peer nodes at time t:

$$P_k(t) = \int_0^t P_{k-1}(x) \cdot \eta_{k-1} e^{-\eta_k(t-x)} dx; \quad P_1(t) = e^{-(N-1)\lambda t}. \tag{2.5}$$

Then, based on Equation (2.5), we derive the average number of copies as a function of time t:

$$E_n(t) = e^{-(N-1)\lambda t} + \sum_{k=2}^{N} k \int_0^t P_{k-1}(x) \cdot \eta_{k-1} e^{-\eta_k(t-x)} dx. \tag{2.6}$$

Figure 2.2 compares the number of copies in the system as a function of time by plotting the results of Equation (2.5), labeled "Markov Chain model"; Equation (2.4), labeled "ODE model"; and the discrete-event simulation. In our simulation model, there are N=50 mobile nodes, in addition to a single sink node. The transmission range of all of the nodes is 25[m]. The network is a closed-square, torus-like area of 1000[m] by 1000[m]. Each node adjusts its velocity independently and maintains the chosen velocity for certain time duration. The time interval between velocity changes is an exponentially distributed random variable with an average of 0.2[sec]. The direction is uniformly distributed in $[0,2\pi]$ and the speed is uniformly distributed in $[20,70]$ [m/s].

9
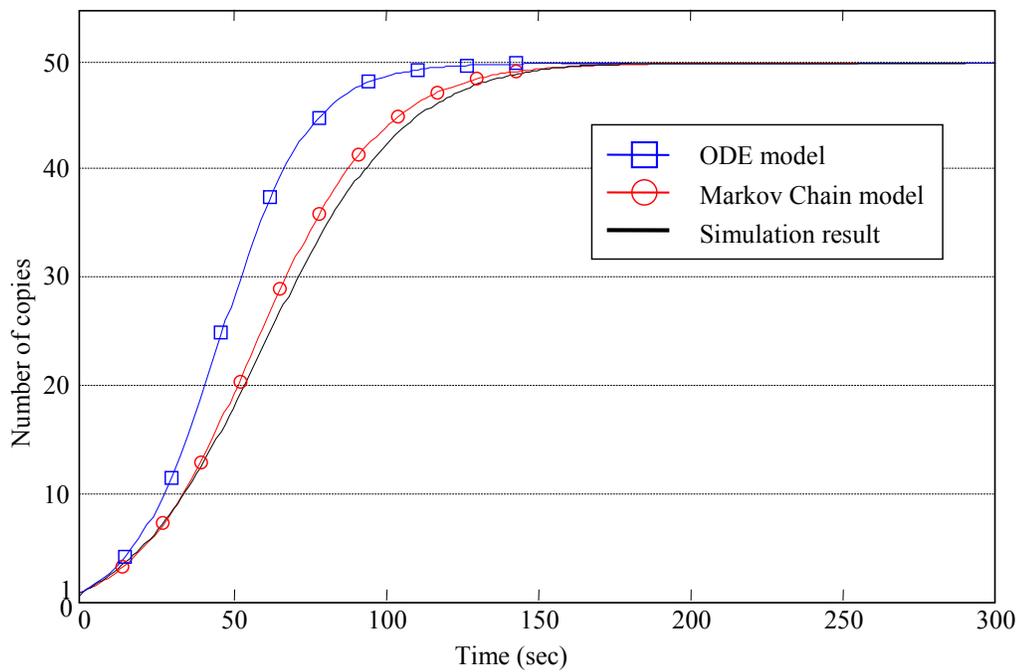
Figure 2.2: Comparison of the number of copies as a function of time

As a result, the encounter rate between any two nodes in our simulation model is, $\lambda = 0.00127$ per second. This random mobility model with its particular encounter rate is used in the rest of this paper.

Figure 2.2 reveals that the Markov Chain model is much closer to the simulation result, compared to the ODE model. The discrepancy that still exists between the Markov Chain model and the simulation result is due to the overlapping of nodes' transmission areas. For example, the case when an infected node encounters two susceptible nodes whose transmission areas overlap, would be treated by the Markov Chain model as two separate encounters with no overlapping area. This results in the probability of such an event being higher than what occurs in the simulation, leading to a small increase in the average number of copies in Figure 2.2 for the Markov Chain case.

2.4. Solution for Epidemic Routing Model

In this work, we consider a single mobile sink node, with the same transmission range and mobility pattern as the other network nodes. Sink nodes can be stationary [2,15], or mobile [38,39]; from the point of view of our analysis, the only difference between these two cases is the encounter rate between the sink node and the other network nodes. With the exception of the sink node, no other network nodes know whether a data packet was received by the sink node. Hence, copies of the packet will remain in the nodes' memories even after the packet is delivered to the sink.

In order to remove these copies of the packet, [15] used the *Time-To-Live* (*TTL*) concept. *TTL*, set by the source node, is a timer which limits the lifetime of a packet in the network. When a packet is transmitted from one node to another, only the residual *TTL* is included in the packet. When the *TTL* expires, all of the replicated packets stored in the network nodes are erased. Although short *TTL* can stop Epidemic

Routing before the number of replicated packets in the system increases excessively, it also reduces the packet delivery probability. Since the number of replicated packets in the system relates to total energy consumption, the initial value of the *TTL* trades off energy consumption for packet delivery probability. The initial *TTL* value can be determined based on the required performance of the Epidemic Routing, as to ensure that the packet reaches the sink before it is erased from the network nodes with some packet delivery probability.

In evaluating the performance of the Epidemic Routing, we consider three metrics: the *number of packet copies in the system*, the *end-to-end delay of the packet*,[3] and the *packet delivery probability*. As depicted in Figures 4 and 5, simulation results show that the number of copies and the packet delivery probability are both non-decreasing functions of time. Hence, a certain value of packet delivery probability corresponds to a certain number of copies at a certain time. For example, the packet delivery probability is 90% at t=122 [sec], at which time the number of copies increases to 45.
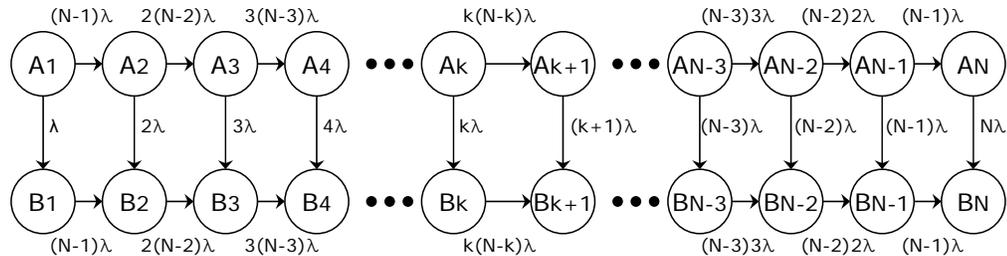
Figure 2.3: The modified Markov Chain Model with states that differentiate based on the packet offload condition

---

[3] Time duration from when the packet is generated until a copy is received by the sink.
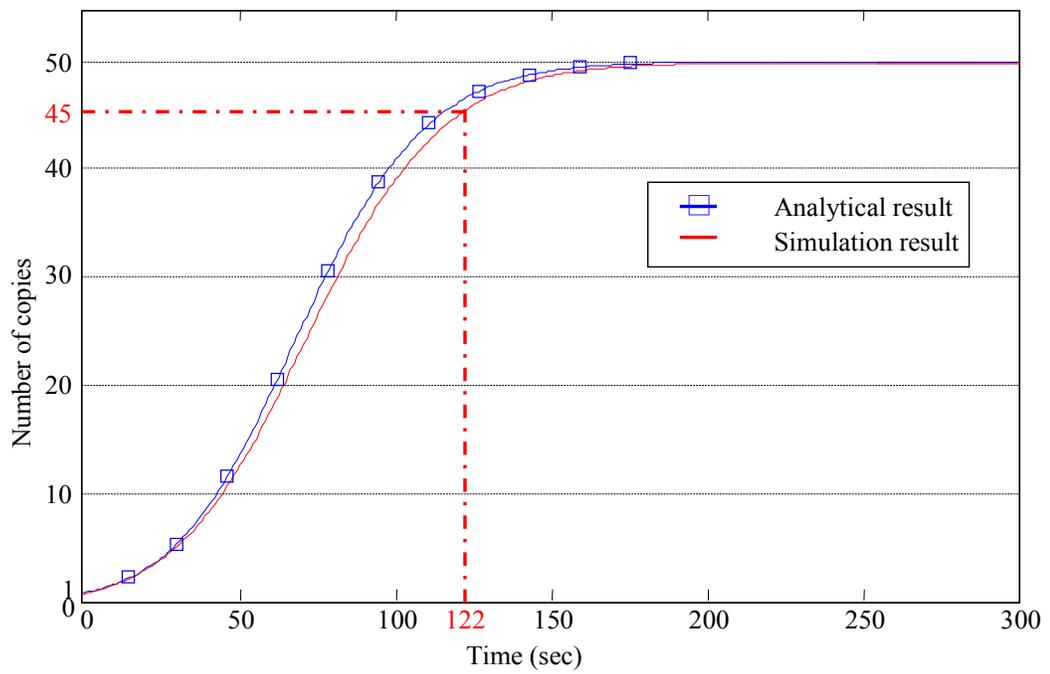
Figure 2.4: Average number of copies as a function of time
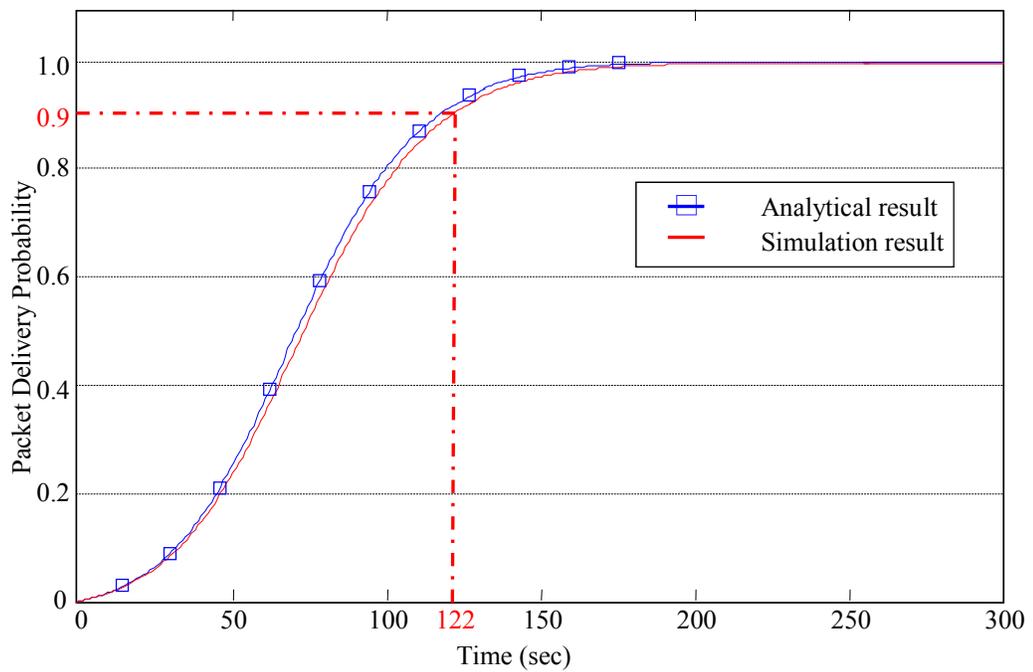
Figure 2.5: Packet delivery probability as a function of time
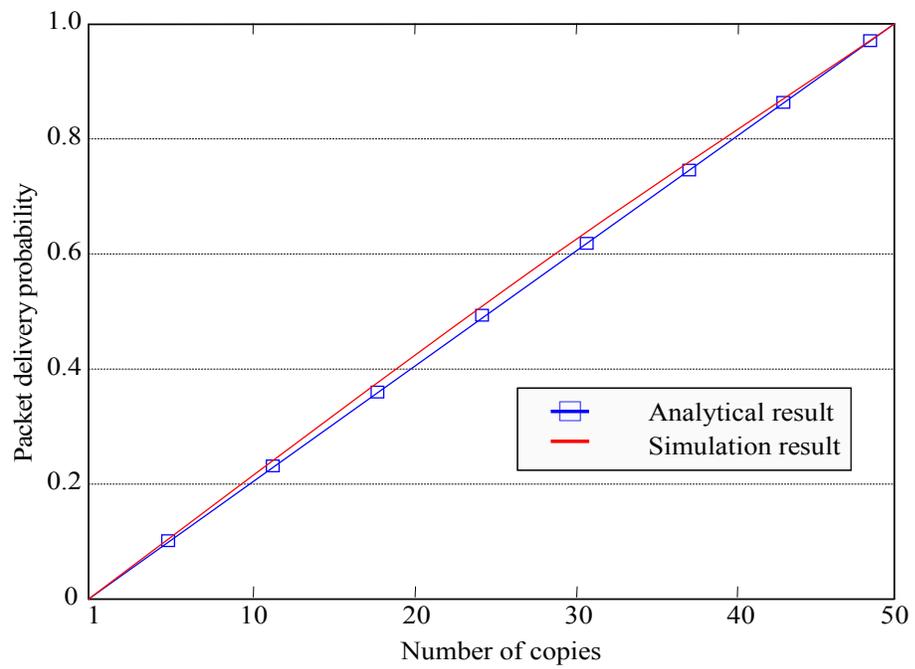
Figure 2.6: Packet delivery probability as a function of number of copies

To differentiate between the cases when, in a particular state of the Markov Chain, the packet has or has not yet been offloaded to the sink, we modify the Markov Chain as shown in Figure 2.3. When the system is in state A_k, there are k copies of the packet in the system, but the sink node has not yet received a copy of the packet. When in state B_k, there are k copies of the packet in the system, but now the sink node has received at least one copy of the packet. Accordingly, the transition rate from state A_k to state B_k is $k\lambda$, which is the encounter rate between any of the k nodes and the sink node.

The probability of having k packet copies in the system at time t, $P_k(t)$, is calculated by using Equation (2.5). From the model in Figure 2.3, we then compute the probability $P_{A,k}(t)$ of the system being in state A_k at time t. The probability of the system being in state B_k at time t, $P_{B,k}(t)$, is obtained by subtracting $P_{A,k}(t)$ from $P_k(t)$:

$$P_{A,k}(t) = \int_0^t P_{A,k-1}(x) \cdot \eta_{k-1} e^{-(\eta_k + k\lambda)(t-x)} dx; \quad P_{A,1}(t) = e^{-N\lambda t}, \tag{2.7}$$

$$P_{B,1}(t) = P_k(t) - P_{A,1}(t). \tag{2.8}$$

Using these probabilities, we derive the packet delivery probability at time t, $D(t)$, as:

$$D(t) = \sum_{k=1}^{N} P_{B,k}(t). \tag{2.9}$$

The expected number of copies of a packet at time t is computed by Equation (2.6). Since the infected nodes do not know whether the sink node received the data packet, the average number of packet copies is independent of packet delivery probability in Equation (2.9). Based on Equations (2.6) and (2.9), the number of copies and the packet delivery probability are plotted as a function of time in Figures 2.4 and 2.5, and further combined in Figure 2.7.
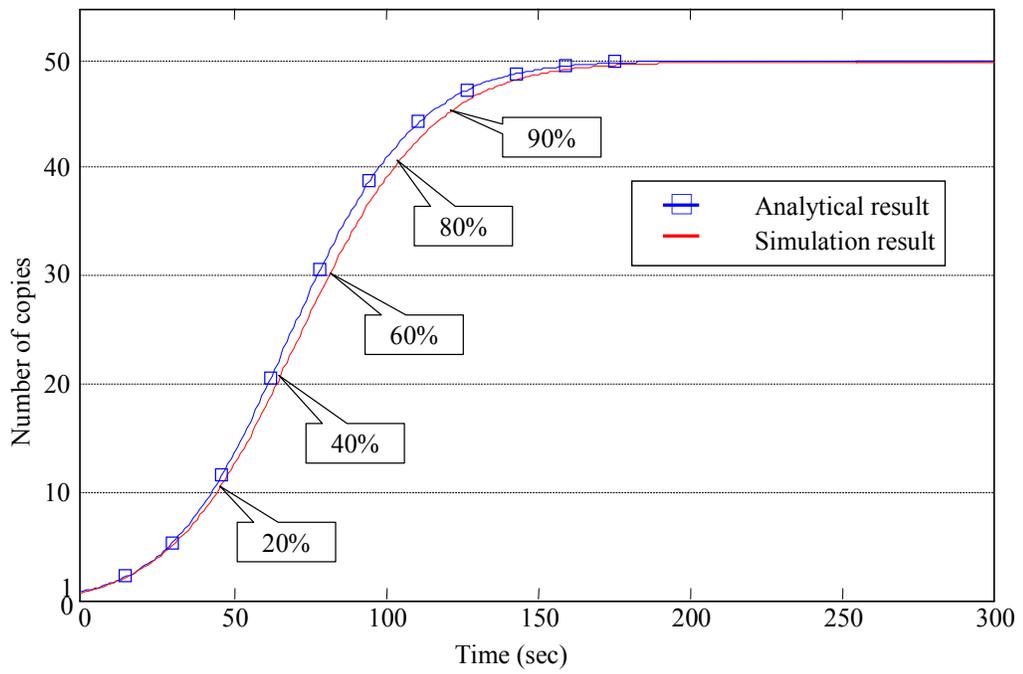
Figure 2.7: Packet delivery probability and the number of copies vs. time

Figure 2.6 shows that packet delivery probability can be considered as a linear function of number of copies. This can be shown using Figure 2.3. Suppose the system is in state A_1. In order for the sink to receive the packet when there is only one copy in the system, the transition from state A_1 to state B_1 should occur before the transition from state A_1 to state A_2. Using the transition rates, the probability of the sink receiving the packet when there is only one copy becomes,

$$R_1 = \Pr[A_1 \rightarrow B_1] = \frac{1}{N}.$$

In order for the sink to receive the packet when there are k copies in the system, the sink should not receive a copy before the number of copies reaches k. Hence, the probability of the sink receiving the packet when there are k number of copies becomes,

$$R_k = \Pr[A_k \rightarrow B_k \mid state = A_k]\Pr[state = A_k] = \frac{1}{N-k+1}\prod_{n=1}^{k-1}\left(1 - \frac{1}{N-n+1}\right) = \frac{1}{N}, \quad (2.10)$$

showing that packet delivery probability is the same for all the states of the system.

2.5.   Energy Consumption

We denote by C_n as the node carrying the nth copy of the data packet; i.e., C_1 is the source node and C_2 is the first node to encounter the source node and receive a copy of the packet. C_3 can be created from either C_1 or C_2. Since encountering either C_1 or C_2 occurs with the same probability, the expected number of transmissions from C_1 to C_3 is 1/2, and the same is true for C_2 to C_3. By the same process, C_n can be created from any of the (n-1) nodes in $\{C_1 \ldots C_{n-1}\}$ and, thus, the expected numbers of transmissions from each node is equivalently 1/(n-1). Hence, for node C_n, the average total number of transmissions required for creation of M copies in the system ($R_{M,n}$) can be derived as:

$$R_{M,n} = \sum_{j=n}^{M-1} \frac{1}{j} \quad (1 \le n \le M-1). \tag{2.11}$$

Using the Markov Chain model from Figure 2.3, we can derive the probability of the sink to be the next to receive a copy of the packet ($S_{N,k}$), when there are k copies in the system and N nodes in the system. When there is only one copy in the system, the probability of the source encountering any peer node is N-1 times larger than the probability of encountering the sink. Thus:

$$S_{N,1} = \frac{1}{N}.$$

When there are k copies, the probability of the source encountering a peer node that does not already have the packet is N-k times the probability of encountering the sink. Hence:

$$S_{N,k} = \frac{1 - \sum_{i=1}^{k-1} S_{N,i}}{N-k+1} = \frac{1}{N} \quad (2 \le k \le N). \tag{2.12}$$

Now suppose there are M copies in the system with a total of N nodes. The probability of the sink receiving a copy when the number of copies reaches n is derived from Equations (2.11) and (2.12). The probability of the sink receiving the packet from any of the nodes in $\{C_1 \dots C_n\}$ is equal. Hence, the expected total number of transmissions by node C_n during this case of the Epidemic Routing ($T_{N,M,n}$) is:

$$T_{N,M,n} = R_{M,n} + \sum_{j=n}^{M} \frac{S_{N,j}}{j} = \sum_{j=n}^{M-1} \frac{N+1}{jN} + \frac{1}{MN} \quad (1 \le n \le M-1), \; T_{N,M,M} = \frac{1}{MN}. \tag{2.13}$$

Obviously, nodes that do not participate in ERP do not consume transmission energy. But Equation (2.13) shows that even for nodes that participate in ERP, energy consumption is not equal. Hence, after multiple packet routings, some nodes will become inoperative due to their battery depletion. Since packet delivery probability is determined by the number of active nodes, packet delivery probability will decrease with time.

CHAPTER 3

Restricted Epidemic Routing

3.1. Restricting the Epidemic Routing

We introduce here a set of variants of ERP, termed Restricted Epidemic Routing (RER), with the purpose of reducing the number of copies of a packet in the system needed to deliver a copy to the sink. Instead of transmitting the data packet to all of the encountered nodes, a node may transmit only to nodes with higher priority, where a node's priority could be based on the history of the node in encountering the sink, the node's mobility pattern, or some other information [40-47]. If the mobility of the nodes is totally random, the priority of the nodes would be equivalent [48-58]. In addition to the priority information, if available, to reduce the number of copies, the control algorithm could rely also on other information. For example, the source node can include a replication restriction in the propagating data packet [15-18]. Of course, restricting packet replication may lead to weaker routing performance, such as an increased delay at a particular level of probability of delivery.

Next, we present three different RER schemes, which restrict the number of copies in ERP. These schemes rely on limiting the number of nodes that participate in packet propagation (the *Exclusion scheme*), limiting the time allowed for propagation (the *Limited Time scheme*), and directly controlling the number of copies (the *Limited Number of Copies scheme*).

3.2. The Exclusion scheme

In the Exclusion scheme (the EX-scheme) some network nodes are prevented from participation in routing the data packet. Before the source propagates the data packet,

20

the source node determines which nodes are excluded from packet propagation in the system. If we label the number of non-excluded nodes as *M,* where $M < N$, the EX-scheme is effectively the ERP, but in a "smaller" network which includes only the *M* non-excluded nodes. Of course, the number of copies in the EX-scheme increases less rapidly than in ERP.

Since the EX-scheme could be seen as simply reducing the total number of network nodes from N to M, the Markov Chain model in Figures 2.3 remains valid, with the modification that N is replaced by M. Consequently, Equations (2.5) and (2.7) remain valid for this case as well, with the simple substitution of *M* for *N*:

$$P_k(t) = \int_0^t P_{k-1}(x) \cdot \mu_{k-1} e^{-\mu_k (t-x)} dx; \quad P_1(t) = e^{-(M-1)\lambda t} \quad \text{and}$$

$$P_{A,k}(t) = \int_0^t P_{A,k-1}(x) \cdot \mu_{k-1} e^{-(\mu_k + k\lambda)(t-x)} dx; \quad P_{A,1}(t) = e^{-M\lambda t},$$

where $\mu_k = k(M-k)\lambda$.

3.3.   The Limited Time scheme

In the Limited Time scheme (the LT-scheme), packets propagate is the same way as in ERP, except that there is a time limit for replication. To accomplish this, another timer is introduced – the *Replication Time Limit (RTL),* which is included within the data packet format, similarly to the *TTL*. The nodes can replicate the data packet until the *RTL* expires, at which time the existing copies persist in the network, albeit without further replication. Of course, $RTL \le TTL$. Assuming that no packet copy was delivered prior to *RTL*, packet delivery probability depends on the number of copies in the system at *RTL*. Replication rates remain the same as in the case of the ERP, and for $0 \le t \le RTL$ the Markov Chain model of the LT-scheme is same as that of the ERP in Figure 2.3. For $t > RTL$, the system follows a simple two-state Markov chain of Figure 3.1, starting either in state B_k, or A_k, depending on whether or not at time $t = RTL$ the

sink received a copy of the packet, respectively, and where k' is the number of copies in the system at time $t = RTL$. Consequently:

for $0 < t \leq RTL$

$$P_k(t) = \int_0^t P_{k-1}(x) \cdot \eta_{k-1} e^{-\eta_k (t-x)} dx; \ \ P_1(t) = e^{-(N-1)\lambda t},$$

$$P_{A,k}(t) = \int_0^t P_{A,k-1}(x) \cdot \eta_{k-1} e^{-(\eta_k + k\lambda)(t-x)} dx; \ \ P_{A,1}(t) = e^{-N\lambda t},$$

and *for* $t > RTL$

$$P_k(t) = P_k(RTL),$$

$$P_{A,k}(t) = P_{A,k}(RTL) \cdot e^{k\lambda(t-RTL)} .$$
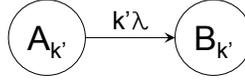
Figure 3.1: The Markov Chain model for the LT-scheme and for $t > RTL$

3.4. The Limited Number of Copies scheme

In the Limited Number of Copies scheme (the LC-scheme), there is an upper limit on the number of copies that the system is allowed to replicate. Of course, disseminating information about the current number of copies in the system to all of the network nodes would be prohibitively expensive in terms of network resources. However, one way to limit the total number of copies is to determine a priori, when the node receives a copy of the packet, how many new copies a node can still create. Another way of looking at this scheme is to assume that upon packet creation all possible replicas are created as well and, from then on, the replicas are just forwarded (typically, as a batch larger than one) to the encountered nodes, but no new replicas are created anymore. A node that stores a single replica can neither forward the replica, nor create any new replicas.

22
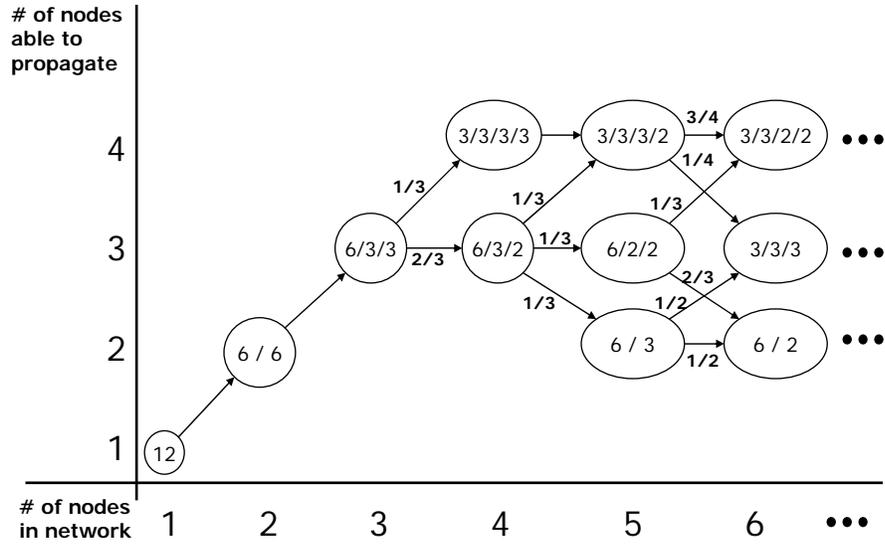
Figure 3.2: A 2D Markov Chain for the LC-scheme with max. of 12 copies

Assume that we want to propagate a packet by up to twelve copies in the system and, upon an encounter, a node carrying more than one copy splits its load with an encountered susceptible node. (When two nodes that already carry one or more replicas of the packet each meet, no exchange takes place.) Figure 3.2 shows the evolution of the system, where numbers in a state indicate the load carried by the replica-carrying (i.e., infected) nodes. For example, in the state [6/2/2], there are five infected nodes: one node with the load of six replicas, two nodes with two replicas, and two other nodes with one replica each. The next system state is determined by the number of nodes that can propagate the replicas, with the assumption that each node has the same probability of encountering another node. Hence, we can calculate the probability of the system being in each state for the 2D Markov Chain in Figure 3.2. For example, when the system is in the [6/3/3] state with 3 infected nodes, and a new node becomes infected, the system will transition to the state [3/3/3/3] with probability of 33.3% and to the state [6/3/2/1] with probability of 66.7%.

In Figure 3.3, n_k is the average number of nodes that can still propagate copies when there are k copies in the system, and m is the maximum number of copies

23

allowed in the system. The n_k can be calculated using the 2D transition Markov Chain; values of n_k are listed in Table 3.1. Using the Markov Chain model in Figure 3,3, the probability of k infected nodes and the probability of k infected nodes without a copy being delivered to the sink are, respectively:

$$P_k(t) = \int_0^t P_{k-1}(x) \cdot \omega_{k-1} e^{-\omega_k (t-x)} dx; \quad P_1(t) = e^{-(N-1)\lambda t} \text{ and}$$

$$P_{A,k}(t) = \int_0^t P_{A,k-1}(x) \cdot \omega_{k-1} e^{-(\omega_k + k\lambda)(t-x)} dx; \quad P_{A,1}(t) = e^{-N\lambda t},$$

where $\omega_k = n_k(N-k)\lambda$. The expected number of copies and the packet delivery probability for a given time t can be calculated by applying the equations above to Equations (2.6) and (2.9).

Due to the limited number of nodes that are able to propagate, the propagation rate in the LC-scheme is slightly smaller than the propagation rate in the LT-scheme, but still larger than the propagation rate in the EX-scheme, since the number of susceptible nodes is not limited.
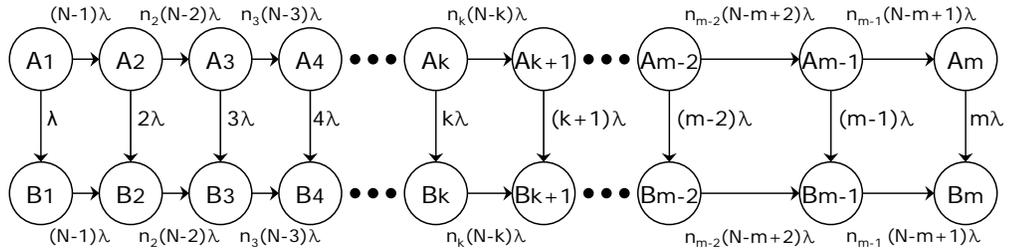
Figure 3.3: Markov chain model for the LC-scheme with max. of m copies

24

Table 3.1: The average number of nodes that can propagate in state k for different values of m in the LC-scheme

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...
m = 2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	⋯
m = 3	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	⋯
m = 4	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0	⋯
m = 5	1	2	1.5	1	0	0	0	0	0	0	0	0	0	0	0	⋯
m = 6	1	2	2	1.5	1	0	0	0	0	0	0	0	0	0	0	⋯
m = 7	1	2	2.5	1.667	1.833	1	0	0	0	0	0	0	0	0	0	⋯
m = 8	1	2	3	2.667	2.333	2	1	0	0	0	0	0	0	0	0	⋯
m = 9	1	2	3	2.833	2.611	2.306	1.75	1	0	0	0	0	0	0	0	⋯
m = 10	1	2	3	3	2.889	2.569	2.278	1.639	1	0	0	0	0	0	0	⋯
m = 11	1	2	3	3.167	3.111	2.921	2.659	2.262	1.631	1	0	0	0	0	0	⋯
m = 12	1	2	3	3.333	3.333	3.231	2.986	2.723	2.262	1.631	1	0	0	0	0	⋯
m = 13	1	2	3	3.5	3.597	3.573	3.403	3.18	2.875	2.261	1.723	1	0	0	0	⋯
m = 14	1	2	3	3.667	3.861	3.894	3.772	3.554	3.344	2.979	2.429	1.874	1	0	0	⋯
m = 15	1	2	3	3.833	4.097	4.179	4.141	4.018	3.772	3.52	3.048	2.496	1.944	1	0	⋯
m = 16	1	2	3	4	4.333	4.444	4.478	4.349	4.132	3.951	3.657	3.104	2.552	2	1	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

3.5.   The Tradeoff Function

In this section, we evaluate and verify the performance of the restricted Epidemic Routing schemes by plotting the average number of copies at time t, $E[n(t)]$. The graphs present results for various degrees of restriction in the RER schemes and were obtained using formulas developed in the previous sections.

Figure 3.4(a) shows an example of the average number of copies in the system as a function of time, with the number of included nodes (M) as a parameter. For larger values of M, the protocol's behavior is closer to ERP and for small M, more like single hop routing. Using Figure 3.4(a) for each M-curve and marking points that correspond to probability of delivering of 95% (triangle) and of 60% (square), we obtain the *Tradeoff function* (the *T-function*) in Figure 3.5.

The T-function of a particular scheme is specific to a particular level of packet delivery probability and presents the tradeoff between the number of copies (which is an indication of energy consumption level) and the packet delivery delay. For the EX-scheme, this tradeoff can be controlled by the value of the parameter M; smaller M leads to a smaller number of copies, but to longer delivery delay and vice versa. Similarly, a corresponding T-function could be obtained for the other two schemes using Figure 3.4(b) and Figure 3.4(c), where the controlling parameters are the value of RTL for the LT-scheme and the maximal number of copies (m) for the LC-scheme, respectively.

Figure 3.6(a) shows the comparison of the T-functions for the three RER schemes and for 90% packet delivery probability. Notice that all of the tradeoff functions start and end at the same point. For any value of time, the T-function of LC-scheme is closer to the origin than the T-functions of the other two schemes. Consequently, we conclude that the LC-scheme outperforms the other two RER schemes in terms of the energy-delay tradeoff.

26
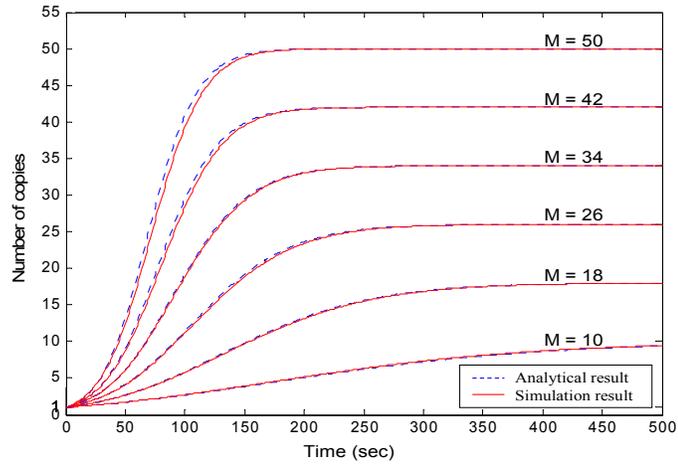
(a) Exclusion (EX) scheme
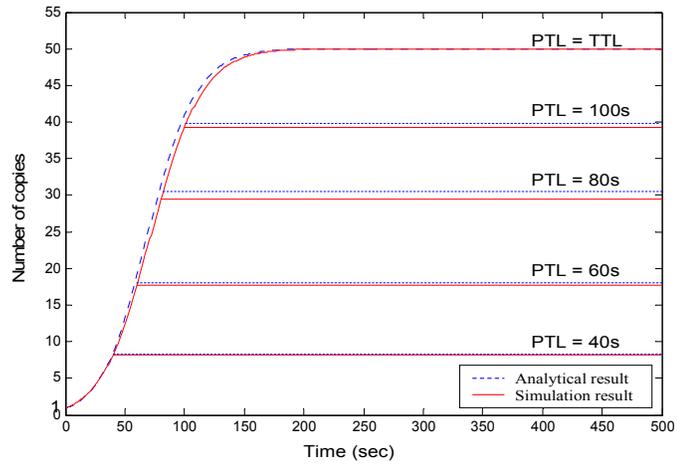
(b) Limited Time (LT) scheme
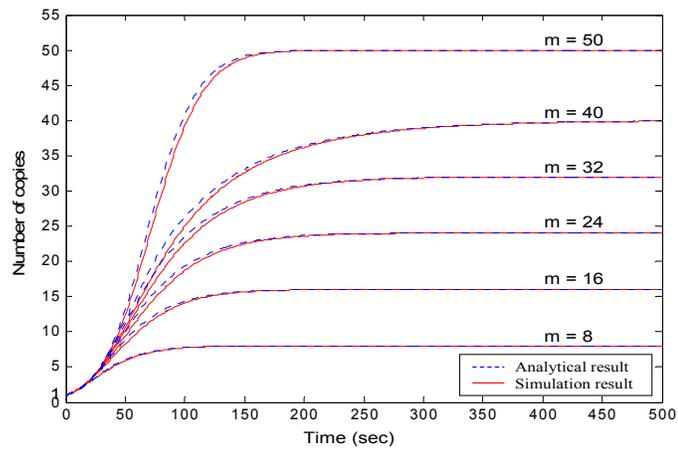
(c) Limited number of Copies (LC) scheme

Figure 3.4: Simulation and analysis results for the RER schemes with varying degrees of restriction
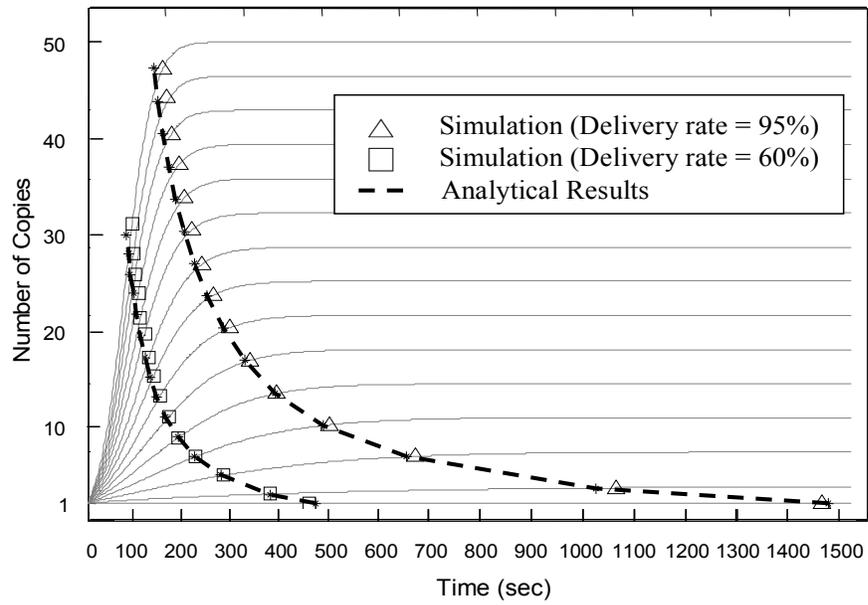
Figure 3.5: The T-function of the EX-scheme for packet delivery probabilities of 60% and 95%
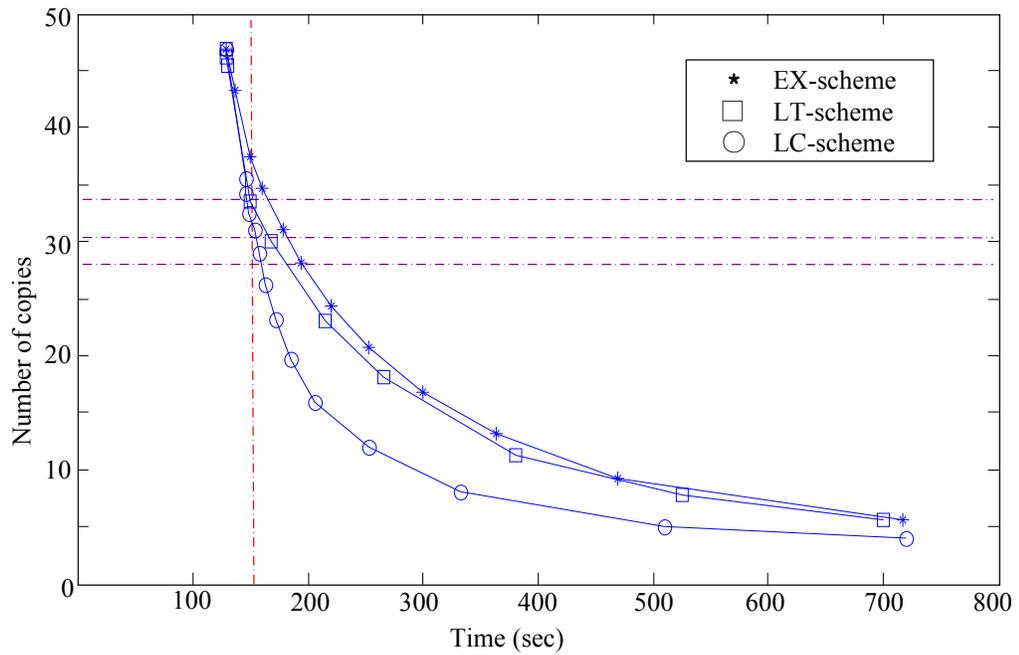
(a) The T-functions at packet delivery probability of 90%
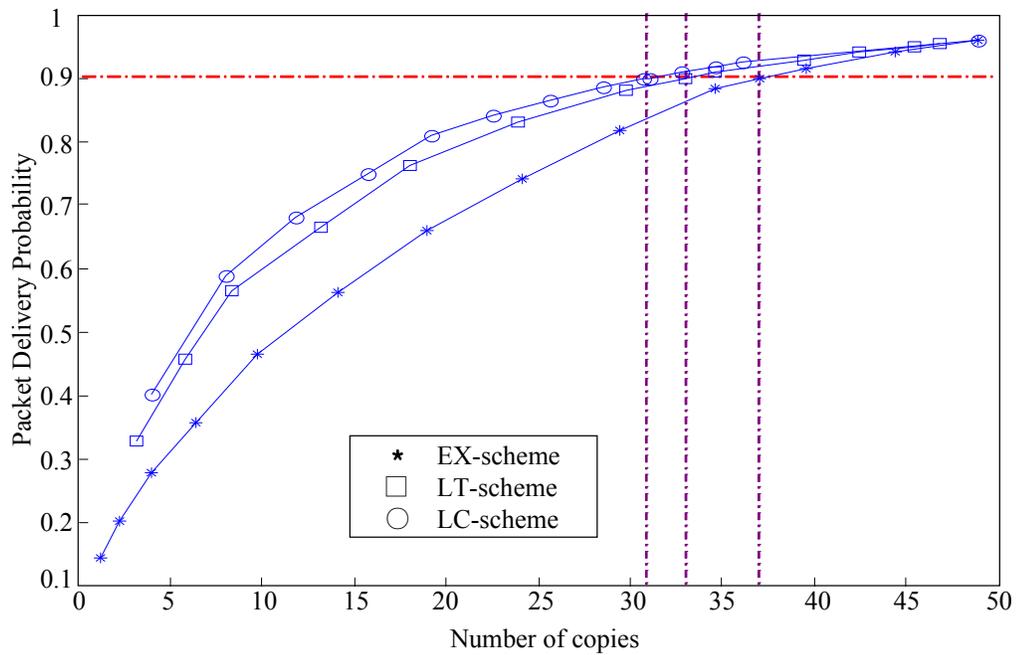
(b) The packet delivery probability at 150 [sec]

Figure 3.6: Comparison of the three RER schemes

Figure 3.6(b) compares the efficiency of the RER schemes in a different way. It shows the packet delivery probability as a function of the number of copies at a fixed time, t=150 [sec]. The figure demonstrates that the LC-scheme has higher packet delivery probability for any number of copies, as compared to the two other schemes. If we set *TTL* to 150 [sec] and fix the packet delivery probability at 90%, the LC-scheme produces an average of approximately 31 copies, while the LT-scheme and EX-scheme produce 33 and 37 copies, respectively.

CHAPTER 4

The Network Lifetime

4.1.   Sequential Packet Routings and the Network Lifetime

In the previous chapter, we proposed and analyzed several RER schemes that conserve battery energy at the network nodes, while maintaining a certain level of packet delivery probability for single packet routing. The main purpose of conserving battery energy is to extend the lifetime of a network that typically would sequentially route many packets through the network. Figure 4.1 depicts the delivery probability of a sequence of 100 consecutive routings in a network using ERP. The network consists of 50 mobile nodes, in addition to one mobile sink node. Energy consumed in one transmission of a copy of a packet is assumed to be 1 energy unit [EU], and we do not account for the energy of packet reception. At the beginning of the network operation, each node is equipped with a battery charged with 40 [EU] and the battery is used for transmissions only. A data packet is created periodically every $T_d = 150$ [sec] at a randomly selected source from among the 50 nodes. The *TTL* is initially set to 150 [sec] in Figure 4.1(a), to 120 [sec] in Figure 4.1(b), and to 75 [sec] in Figure 4.1(c).

4.2.   The MDP Lifetime

The graphs in Figure 4.1 show a gradual decrease in packet delivery probability. Since the batteries of the nodes have limited amounts of energy, nodes with depleted batteries are effectively removed from the network. Thus, as time goes by, the number of active nodes gradually decreases to zero and so does the packet delivery probability. From Figures 4.1 (a) and (b), we can see that the packet delivery probability decreases less rapidly after approaching 20%.

31
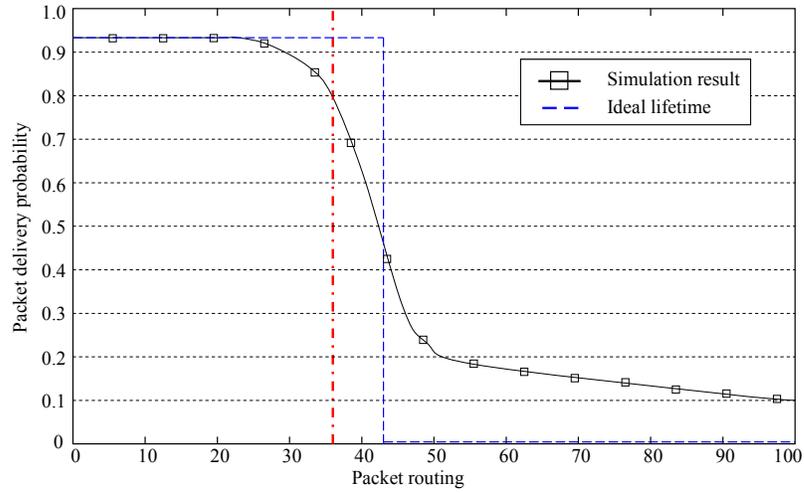
(a) Packet delivery probability at *TTL* = 150 [sec]
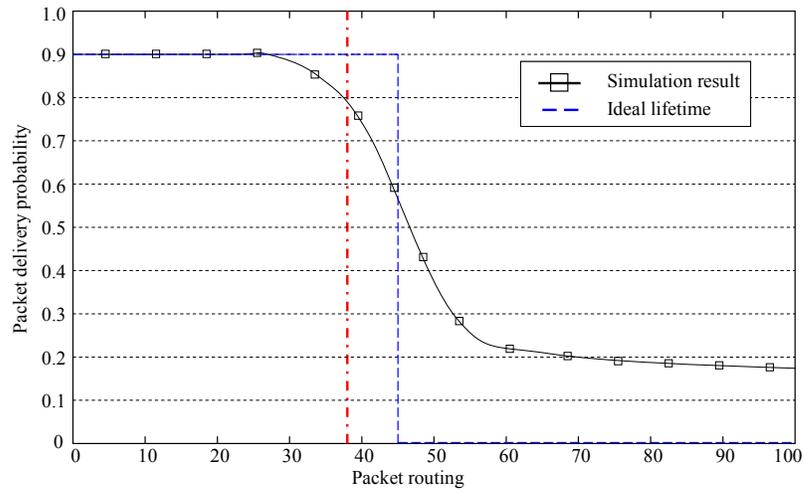
(b) Packet delivery probability at *TTL* = 120 [sec]
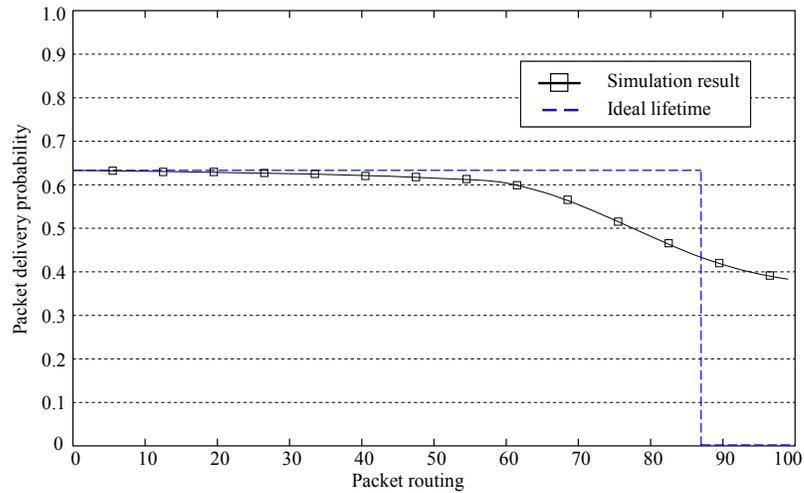
(c) Packet delivery probability at *TTL* = 75 [sec]

Figure 4.1: Packet delivery probability for multiple routings in ER

Since the number of active nodes has decreased, the encounter rate between nodes is also reduced and hence the total energy consumption during single packet routing is significantly lower than in the initial condition. One can postulate that the network continues to be useful only as long as the packet delivery probability is maintained above some level, which we define as the *Minimum Delivery Probability (MDP)*. Consequently, *MDP lifetime* of a network is defined as the time when the packet delivery probability crosses the MDP level from above.

With the MDP set to 80%, MDP lifetime is 36 packet intervals (T_d) when the *TTL* is set to 150 sec. MDP lifetime extends to 39 packet intervals for *TTL* of 120 [sec]. This extension is facilitated by reducing the maximum achievable packet delivery probability when all the nodes are still active. When the value of *TTL* is reduced from 150 [sec] to 120 [sec], the maximum achievable packet delivery probability decreases from 93% to 90%. This example demonstrates the tradeoff between the maximum packet delivery probability and the lifetime of the network; network lifetime can be extended at the expense of reduction in the packet delivery probability. However, if we reduce the *TTL* to 75 [sec], the packet delivery probability never reaches the MDP level, even though it decreases less rapidly as a function of time.

Through appropriate reduction of the value of *TTL*, the MDP lifetime can be extended by decreasing the maximum packet delivery probability to the MDP level. However, this may not always be desirable, as one would often prefer the network to typically achieve the packet delivery probability higher than the MDP level. In other words, typically the network should achieve packet delivery probability of some *Target Delivery Probability (TDP),* where TDP ≥ MDP, for most of its lifetime, and only at the end of the network's lifetime should the packet delivery probability degrade to the MDP level. As we have seen, the MDP lifetime of the network can be extended at the cost of reducing the maximum packet delivery probability to TDP.

Ideally, if the batteries for all of the nodes become depleted at the same time, the packet delivery probability will decrease to zero, as shown by the step-function dotted line in Figure 4.1. For example, in Figure 4.1(a), while the packet delivery probability is maintained at 93%, there were on the average 46.2 packets copied per one routing. Hence, in an ideal case, the duration of the network lifetime should be 50·40 [EU] / 46.2[EU] = 43.29 [routings]. Even after the ideal lifetime has expired, there are still some active nodes participating in delivering packets to the sink. *Ideal lifetime* is achieved by all the network nodes becoming inactive at the same time and can be calculated simply by dividing the total energy in the network by average energy consumed during a packet routing. Ideal lifetime can be considered the maximal extension of the MDP lifetime. Depending on the mobility pattern and geographical position with respect to the sink, nodes consume different amount of energy per particular routing. Consequently, to achieve *ideal lifetime,* sufficient "mixing" of the nodes, as to change their relative positions, is necessary.

4.3. The Network Lifetime of the RER Schemes

In Chapter 3, we have shown the tradeoff function between the number of copies and the packet delivery probability using the three different RER schemes. Using the RER schemes, we can expect that, by reducing the packet delivery probability to TDP, the network lifetime can be extended more efficiently compared to the ca1se in which the network lifetime is extended by reduction in *TTL*. Figure 3.6(b) shows how to obtain the required number of copies at 150 [sec] in each scheme to achieve the TDP of 90%. With these results, we can derive the degree of restriction for each RER scheme that corresponds to this required number of copies at 150 [sec].

The parameters used in this section are the same as those in Chapter 3. In particular, the encounter rate between two nodes is $\lambda = 0.00127$/sec. Additionally, we

define the MDP lifetime by setting TDP to 90% and MDP to 80%. The initial *TTL* is fixed at 150 [sec], and all 50 nodes carry batteries with initial energy of 40 [EU] each. Here, the time interval between packet creations is set to 150 [sec], so that the current packet routing does not influence the next packet routing.

4.4.　The Network Lifetime of the EX-scheme

Figure 3.6(b) shows that the EX-scheme requires 37 nodes in order to obtain the TDP of 90%. Figure 4.2 demonstrates that, when the total number of nodes is limited to $M = 40$, the average number of copies in the system is 37 at 150 [sec]. In order for the EX-scheme to obtain the TDP of 90%, the total number of nodes should be set to 40. Assuming that every node starts with battery of 40 [EU], since the average number of copies at 150 [sec] is 37, the expected lifetime of the EX-scheme is approximately 8108 [sec] (i.e., transmission of $(40 \cdot 50/37) = 54$ data packets, with each packet remaining in the system for 150 [sec]).

4.5.　The Network Lifetime of the LT-scheme

Figure 3.6(b) shows that the LT-scheme requires 33 nodes to achieve the TDP of 90%. As mentioned in Chapter 2, the Markov Chain model of the LT-scheme is basically the same as that of the ERP, hence, we can easily find out from Figure 3.4(b) the time when the number of copies in the system reaches 33. Figure 4.3 demonstrates that, when *RTL* is set to 85 [sec], the average number of copies in the system grows to 33. Assuming that every node has initial battery energy of 40 [EU], since the average number of copies at 150 [sec] for the LT-scheme is 33, the expected lifetime of the LT-scheme is approximately 9091 [sec] (i.e., transmission of $(40 \cdot 50/33) = 60.6$ data packets, with each packet remaining in the system for 150 [sec]).
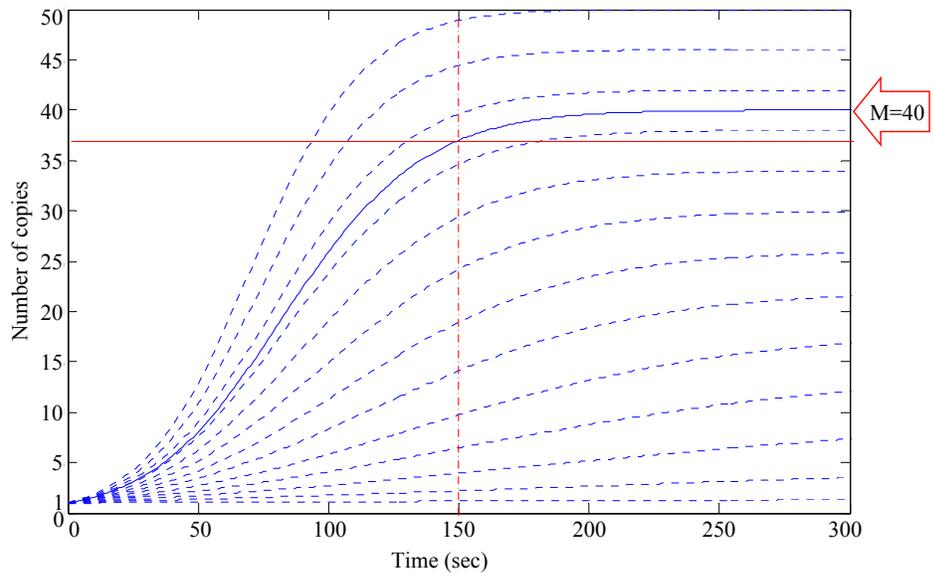
Figure 4.2: The performance of the EX-scheme - 37 copies at 150 [sec]
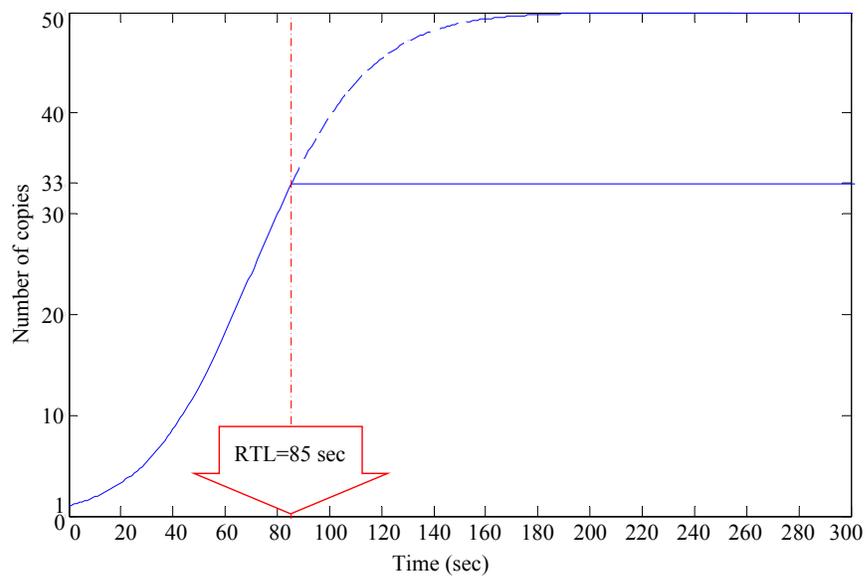
Figure 4.3: The performance of the LT-scheme - 33 copies at 150 [sec]
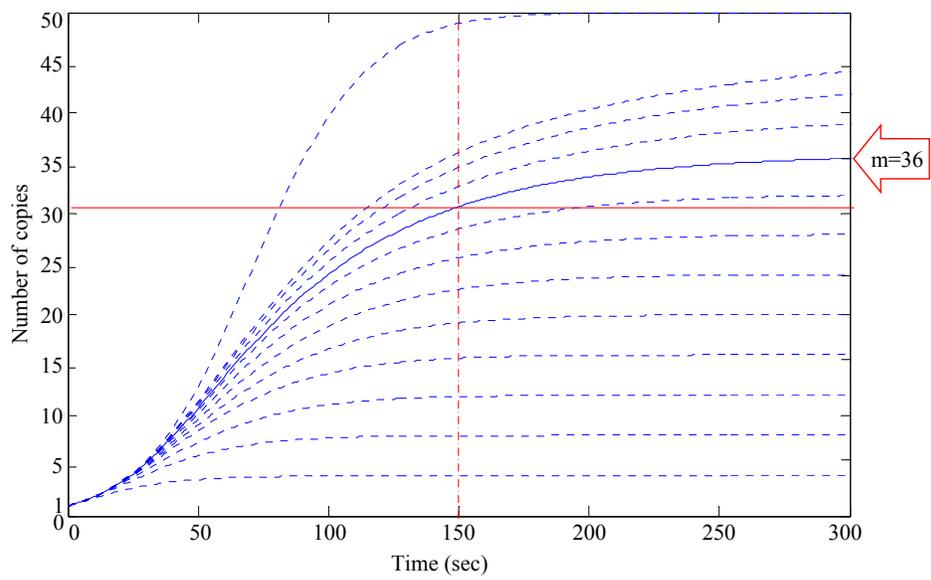
Figure 4.4: The performance of the LC-scheme - 31 copies at 150 [sec]

4.6. The Network Lifetime of the LC-scheme

Figure 3.6(b) shows that the LC-scheme requires 31 nodes in order to obtain the TDP of 90%. Figure 4.4 demonstrates that, when the total number of copies is limited to $m = 36$, the average number of copies in the system grows to 31 at 150 [sec]. Hence, in order for the LC-scheme to satisfy the TDP of 90%, the total number of copies should be upper-bounded by 36. Assuming that every node has initial battery energy of 40 [EU], the expected lifetime of the LC-scheme is approximately 9677 [sec] (i.e., transmission of $(40 \cdot 50/31) = 64.5$ data packets, with each packet remaining in the system for 150 [sec]). Notice that the RER scheme with the longest expected lifetime is the LC-scheme, which was shown to have the most efficient T-function in Chapter 3.

4.7. The Residual–Energy Information

Performance of the Epidemic Routing scheme depends heavily on the number of active nodes in the system. To obtain the *ideal lifetime*, all of the nodes' batteries should be depleted at the same time. Of course, without a special mechanism implemented in the network, this is highly improbable to happen. However, using the residual energy information, it is possible to implement a control algorithm to manage the energy consumption of each node. Such a scheme requires each node to be aware of its residual energy and transmit this information to the encountered node prior to transmitting any data packet.

Suppose that at the encounter time of two nodes, the nodes share their residual energy information. A prudent thing to do is to allow the node with larger residual energy to create more copies in future encounters, as compared to the node with less residual energy. Furthermore, since the LC-scheme is already designed to allow controlling the number of copies that a node could spin off in its future encounters, it

would suggest that the LC-scheme, combined with the above energy control mechanism, could result in more uniform energy depletion among network nodes.

4.8.   The LC-scheme with Residual–Energy Information

Normally, in the LC-scheme, when a node transmits a copy to another node, it divides its load (number of copies that could be created in future encounters) and passes half of that load on to the receiving node. That way, the two nodes end up with about the same number of copies to propagate. However, if the nodes share their residual energy information, they should divide the number of copies according to residual battery energy, instead. The simplest way to divide the load is to split the number of copies in proportion to residual energies of the nodes. We refer to this scheme as the *LE-scheme*. In order to find the parameter m that limits the total number of copies for the LE-scheme, one could use the same method as for the LC-scheme.

4.9.   Other RER schemes with Residual–Energy Information

Although not impossible, the method that we used to extend the LC-scheme to the LE-scheme is difficult to apply to the EX and the LT schemes, since these two schemes cannot naturally control the number of transmissions of nodes in the future. To clarify this further, suppose that we extend the EX- or the LT-scheme by allowing a node to transmit to another node only when the first node has more residual battery energy than the second node. If, for instance, the source node has less residual battery energy than any node that it encounters, then the packet can be still propagated in the LE-scheme. However, with the EX-scheme or the LT-scheme, the only opportunity for the packet to reach the sink would be for the source to deliver the packet to the sink itself. Even if we allow a node to transmit based on some probabilistic function of its residual energy, this would reduces the propagation rate, while decreasing the

40
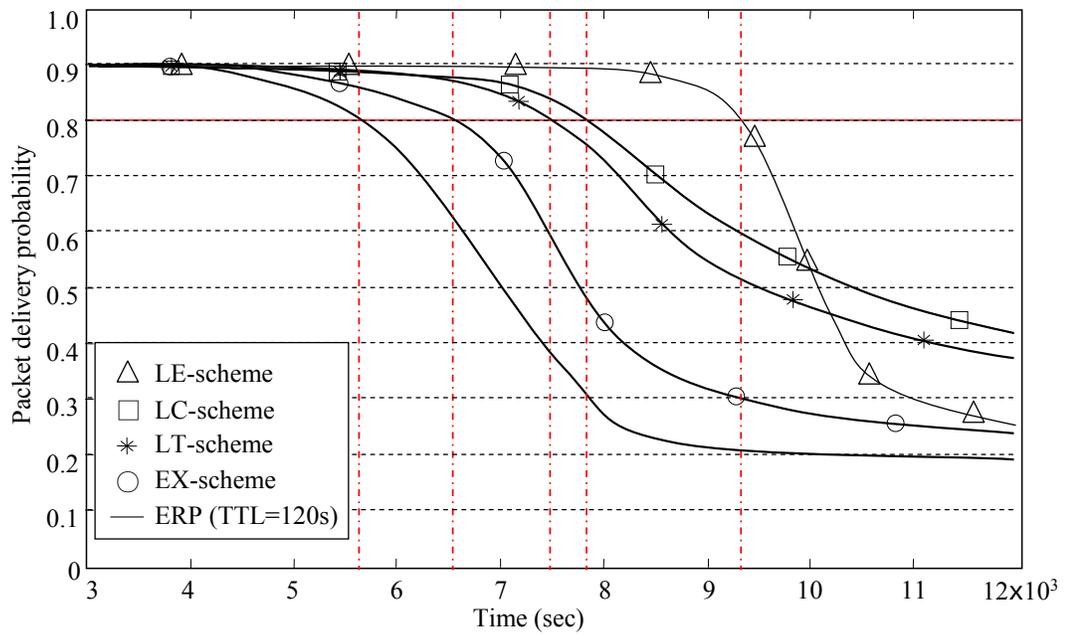
Figure 4.5: Comparison of Lifetime at *MDP* = 80%

Table 4.1: The lifetime of the ERP and RER schemes

Scheme	Ideal lifetime (α)	MDP lifetime (β)	α-β	E(n)	σ(n)	σ(n) /E(n)
ERP	6667 s	5850 s	817 s	0.90	1.360	1.511
EX	8108 s	6450 s	1658 s	0.74	1.206	1.630
LT	9091 s	7500 s	1591 s	0.66	1.187	1.798
LC	9677 s	7800 s	1877 s	0.62	1.213	1.956
LE	9677 s	9250 s	427 s	0.62	0.904	1.458

packet delivery probability below TDP. In order to satisfy TDP, the number of copies would have to be increased, reducing (or possibly eliminating) the effectiveness gained by extending the scheme to rely on the residual energy information.

4.10. Comparison of the Lifetime Performance

Figure 4.5 shows a comparison of the 80% MDP lifetimes for all of the schemes we have discussed, including the ERP, and for our example network scenario. After approximately 4×10^3 sec, the packet delivery probability of the ERP scheme starts to decrease and the rate drops below the MDP at approximately 5850 [sec]. One can see that the EX-scheme has the shortest MDP lifetime, while the LE-scheme has the longest MDP lifetime, from among the RER schemes. The LE-scheme has a longer MDP lifetime than other RER schemes, since it requires the smallest number of copies.

The MDP lifetimes for the various schemes are summarized in Table 4.1 and compared with the corresponding expected ideal lifetimes. *E(n)* is the average number of transmissions per node per routing and *σ(n)* is the standard deviation of the number of transmissions per node per routing for the packet delivery probability of TDP. The last column is the coefficient of variation values calculated as *σ(n)/E(n)*.

Indeed, the results in Table 4.1 demonstrate again that the LE-scheme achieves the longest MDP lifetime. Although the other RER schemes have longer MDP lifetimes than that of the ERP, the difference between the ideal lifetime and the MDP lifetime is larger for the EX-, the LT-, and the LC-scheme, compared to this difference for the ERP scheme. The results in the table confirm our postulate that a large value of the coefficient of variation of the average number of transmissions per node suggests a large difference between the MDP lifetime and the ideal lifetime. Using residual energy information, the variance of energy consumption per node can be reduced, which also reduces the difference between the MPD lifetime and the ideal lifetime.

43

CHAPTER 5

Epidemic Routing with Limited Memory

Extending the network lifetime is typically based on the tradeoff between the energy consumption and the packet delivery probability only when the packet delivery probability under ERP exceeds the TDP. In this chapter we will consider the case when the packet delivery probability is below the TDP.

5.1.   Sequential ERP with Limited Node Memory

In the sequential ERP with limited node memory, a node cannot always receive packet from an encountered node and store it, even if the packet is not in its memory. Due to the limited memory, each node has to set a priority of determining which packet to store and which packet to remove from its memory. Suppose node A encounters node B while its memory is full. If node B carries a packet which takes priority over some of the packets in node A (in other words, node B is an infected node and node A is a susceptible node of the packet), node A will remove the packet with lowest priority from its memory and replace it with the packet from node B. This requires an exchange of a short ACK packet including the priority information of carrying packets before the actual packet transmission. We will assume that this ACK is much smaller than the actual data packet and, hence, overlook the time duration and energy consumption for the ACK exchange.

Suppose that a packet is generated with a sequence number k, and the packet with the larger sequence number has higher priority. In other words, when a node memory is full, the oldest packet is replaced by a newer packet. Before the next packet is generated, packet P_k will be routed following the basic ERP since the nodes carrying

the previous packets can be considered as susceptible nodes of P_k. Obviously the number of susceptible nodes will decrease as the number of infected nodes increases. As the network starts to route the subsequent packets, P_{k+1}, P_{k+2}, P_{k+3} …, the number of infected nodes of P_k will also decrease due to the increased number of copies of the subsequent packets. Eventually the number of P_k in the system will become zero.

5.2. Fundamental model

The sequential ERP model with limited memory operates in different ways based on the generation of subsequent packets. When P_1 is generated from the source, the model is identical to Figure 5.1, which is the same model as the basic ERP model with unlimited node memory, until the next packet P_2 is generated. When the network generates a new packet, we need to switch to a new model. To see how limited memory affects the ERP, we will assume an extreme case in which each node can store only one packet in its memory. In our model, there is one source generating packets, and one sink which has sufficiently large memory to store received packets.
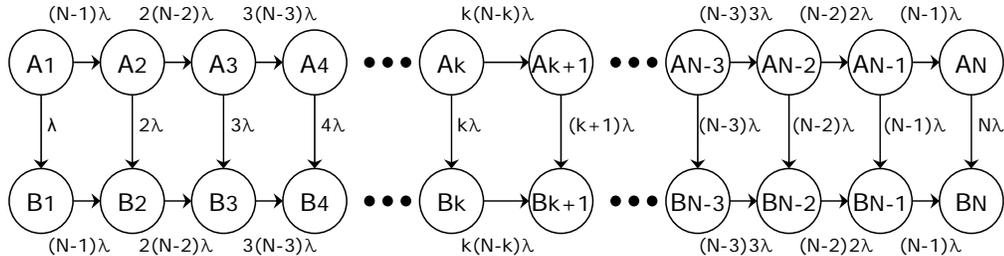
Figure 5.1: Model of limited memory ERP without subsequent packets

Suppose the number of copies of P_1 is x. From the point of view of P_1, the subsequent packets, P_2, P_3, P_4 …, can be considered as a single packet. Hence, the total number of subsequent packets can be stated as a single variable y. Our new model, Figure 5.2, is a modification of Figure 5.1. Each state A_k is replaced with a set of states $\{A_{x,y} \mid x=k, 0 \le y \le N-x\}$ where the subscripts k and x indicate the number of

P_1, and subscript y indicates the total number of subsequent packets. Note that the sum $(x+y)$ is limited to the total number of nodes N. While the system is routing only P_1, the system is in the states in which $y=0$ until the source generates P_2.

If a node carrying P_1 encounters a susceptible node, then x increases by one while y stays constant. If a node carrying one of the subsequent packets encounters a node susceptible to both P_1 and the subsequent packets, then y increases by one while x stays constant. If a node carrying P_1 encounters a node carrying a subsequent packet, then x decreases by one and y increases by one, since P_1 is replaced by the subsequent packet. As we can see in Figure 5.2, these cases requires additional transitions between different states, compared to Figure 5.1, and they occur at different rates dependent on the current number of packets x and y. This also applies for the set of states $\{B_{x,y} \mid x=k, 0 \leq y \leq N-x\}$. When P_1 arrives at the sink, there is a transition from state $A_{x,y}$ to state $B_{x,y}$ with transition rate $x\lambda$. When x reaches 0, P_1 is permanently removed from the system. This creates a 3D transition Markov chain model with absorbing states $A_{0,N}$ and $B_{0,N}$.
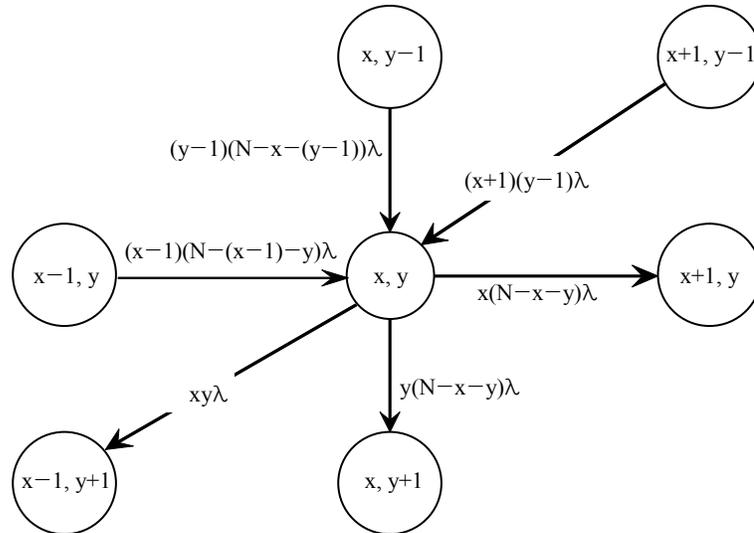
Figure 5.6: Model of limited memory ERP with subsequent packets

46

5.3. Solving the Limited Memory ERP with Subsequent Packets

Suppose the time interval between packet generations in our model is T_d. For time $t < T_d$, based on Figure 5.1, the number of copies and the packet delivery probability of P_1 can be derived from the equations used in Chapter 2. For time $t \geq T_d$, we first need to derive the probability of being in an initial state, either $A_{x,y}$ or $B_{x,y}$, at T_d, and then use the new 3D transition Markov chain model to derive the probability of being in a state, $A_{x,y}$ or $B_{x,y}$, for time $t > T_d$. These probabilities can be derived using the following steps.

Step 1: Since the source generates a new packet P_2 and removes P_1 from its memory at T_d, x decreases by one and $y=1$. Hence, if the final state of Figure 5.1 is A_{k+1} or B_{k+1} at T_d, then the initial state of Figure 5.2 is $A_{k,1}$ or $B_{k,1}$.

For Figure 5.1, suppose that the probability of being in state A_k at T_d is $\alpha(k)$ and the probability of being in sate B_k at T_d is $\beta(k)$. For Figure 5.2, suppose that the probability of being in state $A_{x,y}$ for $t > T_d$ is $\alpha_{x,y}(t)$ and the probability of being in state $B_{x,y}$ for $t > T_d$ is $\beta_{x,y}(t)$. Then the initial probability $\alpha_{k,1}(T_d) = \alpha(k+1)$ and $\beta_{k,1}(T_d) = \beta(k+1)$, where $0 \leq k \leq N-1$.

Step 2: Using the transition rates between different states from Figure 5.2, we first derive the following differential equations in order to derive the probabilities $\alpha_{x,y}(t)$ and $\beta_{x,y}(t)$ for $t \geq T_d$.

For $t \geq T_d$, let $\tau = t - T_d$,

$$G_1(x, y) = x(N - x - y)\lambda, \ \ G_2(x, y) = y(N - x - y)\lambda, \ \ G_3(x, y) = xy\lambda, \ \ G_4(x) = x\lambda.$$

For $x = 0$ and $y = 1$,

$$\frac{d\alpha_{0,1}(\tau)}{d\tau} = -\alpha_{0,1}(\tau) \cdot G_2(0,1), \quad \frac{d\beta_{0,1}(\tau)}{d\tau} = -\beta_{0,1}(\tau) \cdot G_2(0,1).$$

For $x = 0$ and $2 \le y \le N$,

$$\frac{d\alpha_{0,y}(\tau)}{d\tau} = \alpha_{0,y-1}(\tau) \cdot G_2(0, y-1) + \alpha_{1,y-1}(\tau) \cdot G_3(1, y-1) - \alpha_{0,y}(\tau) \cdot G_2(0, y),$$

$$\frac{d\beta_{0,y}(\tau)}{d\tau} = \beta_{0,y-1}(\tau) \cdot G_2(0, y-1) + \beta_{1,y-1}(\tau) \cdot G_3(1, y-1) - \beta_{0,y}(\tau) \cdot G_2(0, y).$$

For $1 \le x \le N-1$ and $y = 1$,

$$\frac{d\alpha_{x,1}(\tau)}{d\tau} = \alpha_{x-1,1}(\tau) \cdot G_1(x-1,1) - \alpha_{x,1}(\tau) \cdot (G_1(x,1) + G_2(x,1) + G_3(x,1) + G_4(x)),$$

$$\frac{d\beta_{x,1}(\tau)}{d\tau} = \alpha_{x,1}(\tau) \cdot G_4(x) + \beta_{x-1,1}(\tau) \cdot G_1(x-1,1) - \beta_{x,1}(\tau) \cdot (G_1(x,1) + G_2(x,1) + G_3(x,1)).$$

For $1 \le x \le N-1$ and $2 \le y \le N$,

$$\frac{d\alpha_{x,y}(\tau)}{d\tau} = \alpha_{x+1,y-1}(\tau) \cdot G_3(x+1, y-1) + \alpha_{x-1,y}(\tau) \cdot G_1(x-1, y) + \alpha_{x,y-1}(\tau) \cdot G_2(x, y-1)$$
$$- \alpha_{x,y}(\tau) \cdot \left(G_1(x, y) + G_2(x, y) + G_3(x, y) + G_4(x) \right),$$

$$\frac{d\beta_{x,y}(\tau)}{d\tau} = \alpha_{x,y}(\tau) \cdot G_4(x) + \beta_{x+1,y-1}(\tau) \cdot G_3(x+1, y-1) + \beta_{x-1,y}(\tau) \cdot G_1(x-1, y)$$
$$+ \beta_{x,y-1}(\tau) \cdot G_2(x, y-1) - \beta_{x,y}(\tau) \cdot \left(G_1(x, y) + G_2(x, y) + G_3(x, y) \right).$$

Step 3: Using the equations from step 1 and step 2, we can derive the average number of copies $E_n(t)$, the packet delivery probability $D(t)$, and the total number of transmission $Tr(t)$ of P_1 for $t \ge T_d$ as follows:

$$\alpha_{x,y}(t) = \alpha_{x,y}(T_d) + \int_0^{t-T_d} \frac{d\alpha_{x,y}(\tau)}{d\tau} d\tau,$$

$$\beta_{x,y}(t) = \beta_{x,y}(T_d) + \int_0^{t-T_d} \frac{d\beta_{x,y}(\tau)}{d\tau} d\tau,$$

$$E_n(t) = \sum_{x=1}^{N-1} \sum_{y=1}^{N-x} x \cdot \left(\alpha_{x,y}(t) + \beta_{x,y}(t) \right), \tag{5.1}$$

$$D(t) = \sum_{x=0}^{N-1} \sum_{y=1}^{N-x} \beta_{x,y}(t), \tag{5.2}$$

$$\frac{dTr(\tau)}{d\tau} = \sum_{x=0}^{N-1}\sum_{y=1}^{N-x}\left(\alpha_{x,y}(\tau) + \beta_{x,y}(\tau)\right)\cdot G_1(x,y),$$

$$Tr(t) = \sum_{k=0}^{N-1}\left(k\left(\alpha_{k,1}(T_d) + \beta_{k,1}(T_d)\right)\right) + \int_0^{t-T_d}\frac{dTr(\tau)}{d\tau}\,d\tau. \qquad (5.3)$$

The solid lines plotted in Figure 5.3 show the analytical results based on Equations (5.1)−(5.3) with different time intervals between packet generations. The analytical results are compared with the simulation results, which are plotted in dotted lines. In our simulation model, there are N=50 mobile nodes, in addition to one sink node. The transmission range of all of the nodes is 25[m]. The network is a closed square of 1000[m] by 1000[m] torus-like area. Each node determines its velocity independently and, after maintaining the velocity for a certain time, changes its speed and direction. The time duration between velocity changes is an exponentially distributed random variable with an average of 0.2sec. The direction is uniformly distributed in [0, 2π] and the speed is uniformly distributed in [20, 70] [m/s]. As a result, the encounter rate in our simulation model is, $\lambda = 0.00127$ /sec.

As we can see in Figure 5.3(a), the number of copies increases for a while and then decreases to zero somewhat symmetrically. As the time interval, T_d, increases, the time when the number of copies becomes zero increases as well. The maximum number of copies also increases, but it is bounded by the total number of nodes. As the time interval T_d goes to infinity, the number of copies will be similar to the ERP with unlimited node memory.

Since the number of copies becomes zero at some point, the total number of transmissions stops increasing and converges to a certain limit, as shown in Figure 5.3(b). As the time interval T_d increases, the limit increases as well, but it is bounded by a maximum value (*N-1*), where N is the total number of nodes in the system. For the same reason, the packet delivery probability stops increasing and converges to a certain limit. Figure 5.3(c) shows that, as the time interval T_d increases, the limit
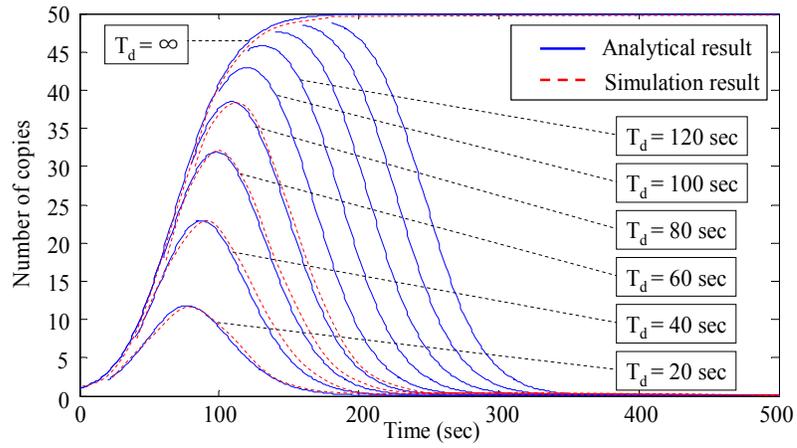
(a) Number of copies for different T_d
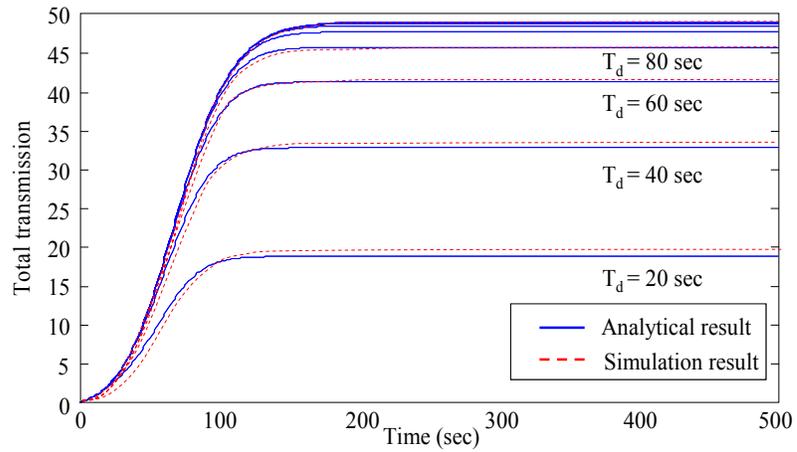
(b) Total transmission for different T_d
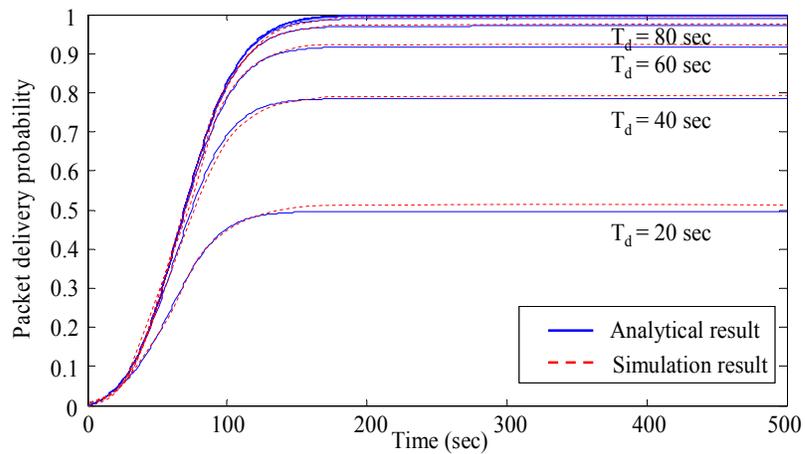
(c) Packet delivery probability for different T_d

Figure 5.7: Results for ERP with limited node memory

50

increases, and it is bounded by a maximum probability of 1. We can conclude that, when routing a packet under ERP with limited node memory, the packet delivery probability can be affected by the packet generation rate. For example, when the source in our simulation is generates 0.05 packets/second (T_d = 20 sec), the packet delivery probability is approximately 50%.

5.4.   Threshold of Packet Generation Rate

As we have seen from the previous section, the packet delivery probability for P_1 converges to a certain limit. This value can be 100% or smaller depending on the packet generation rate. This is also the case for subsequent packets as long as they are generated sequentially with a fixed generation rate. Even when the source randomly generate packets at any time, it is still possible to fix the time interval T_d by using fixed time slots to determine the start point of each packet routing. As long as the source generates a large number of packets ($>> k$), the routing process of P_k (k > 1) is no different from P_1, expectation of the packet delivery probability should be the same. Practically, we can observe the packet delivery probability for sequential packets converging to the same value. We will call this converging value the Sequential Delivery Probability (SDP).

Suppose we set the Target Delivery Probability (TDP) at 92%, which is the SDP for T_d = 60 sec. Based on Figure 5.3(c), in order to have an SDP larger than the TDP, the time interval between each packet generation should be longer than T_d = 60 sec. More specifically, the packet generation rate, μ, should be $\mu < 0.0167$ packets/sec where 0.0167 is the threshold value of the packet generation rate. We will define this range $\mu < 0.0167$ packets/sec as the *Low Packet Generation Rate*. Figure 5.4(a) shows an example of the SDP for a low packet generation rate where $\mu = 0.01$ packets/sec. We can see that the SDP converges to approximately 98%.
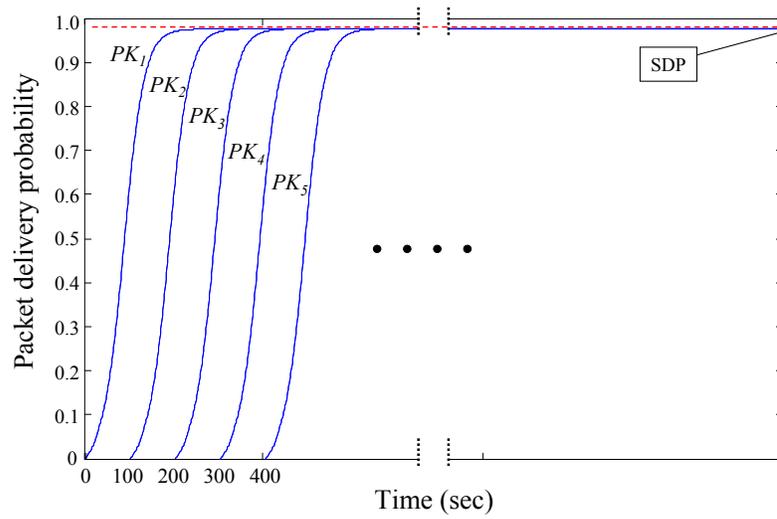
(a) Low generation rate
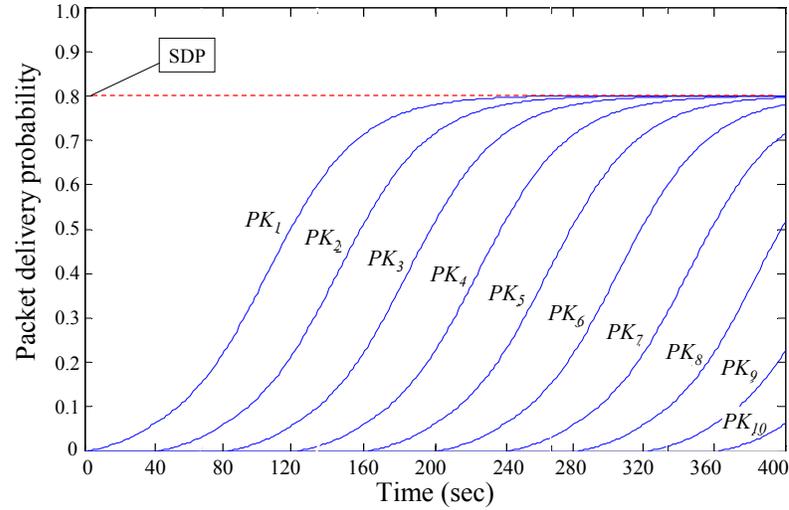
(b) High generation rate

Figure 5.8: Sequential Delivery Probability for fixed packet generation rate

Since the SDP exceeds the TDP, the low packet generation rate case is similar to the basic ERP with unlimited node memory. Hence, it is possible to extend network lifetime at the expense of reducing the SDP to the TDP using Restricted Epidemic Routing schemes we proposed in Chapter 3.

If the packet generation rate is high, each packet will not have enough time to be delivered at the sink without being removed from the memory. As shown in Figure 5.3(c), if $\mu > 0.0167$ packets/sec, then the SDP is lower than the TDP. We will define this range $\mu > 0.0167$ packets/sec as the *High Packet Generation Rate*. Figure 5.4(b) shows an example of the SDP for a high packet generation rate where $\mu = 0.025$ packets/sec. We can see that the SDP converges to approximately 80% which is 12% lower than the TDP.

Routing protocol that cannot achieve the TDP is obviously useless. We can still achieve TDP by blocking a portion of packets and maintain the threshold generation rate. However, the source has to be aware of the SDP in order to determine which packets to block. Even though the source is aware of the SDP and is able to determine which packets to block, the overall packet delivery probability does not get improved.

In Figure 5.4(b), for example, if the source blocks every one out of three packets and controls the packet generation rate at 0.0167 packets/sec ($T_d = 60$ sec), this will result in a 92% packet delivery probability with a loss of 30% of the packets. On average, the actual packet delivery rate will be 61% which is worse than 80%. In order to achieve the TDP without any loss, we propose a new protocol in the next chapter.

CHAPTER 6

Linear Network Coding Applied ERP

6.1.   Linear Combination

We first assume that all network nodes, including the source and sink nodes, are capable of processing Linear Network Coding (LNC). Using LNC, a node may linearly combine two or more consecutive packets. Our Network Coding applied ERP (NC-ERP) operates as follows.

The initial packet P_1 is routed in the network under the basic ERP. For subsequent packets, when a new packet P_k is generated, the source creates a combination packet, C_k, referred to here as the kth combination, and the value of k is called the epoch. C_k is a linear combination of the packets P_k and P_{k-1}:

$$C_k = \alpha_0 P_k + \alpha_1 P_{k-1}, \tag{6.1}$$

where α_0 and α_1 are coding coefficients randomly chosen from a Galois field. Each combined packet contains the combination vector $[k, \alpha_0, \alpha_1]$ in its header. We define G_k as the group of nodes that carry a combination C_k.

The source creates and transmits a different combination every time it encounters another node which is not in G_k. When two nodes come into contact with each other, they first exchange their combination vector through an ACK. If the two combination vectors are identical, no further transmission occurs. However, when the value of the epoch is different at the two nodes, say k and $k-1$, the node carrying the combination C_{k-1} removes C_{k-1} from its memory to store C_k. If a node carrying a combination C_k encounters a node that does not carry any combination, the combination C_k is copied onto the empty node.

Suppose now that two nodes that belong to the same G_k, but carry different combinations C_k^0 and C_k^1 come into contact, where $C_k^0 = \alpha_{00}P_k + \alpha_{01}P_{k-1}$ and $C_k^1 = \alpha_{10}P_k + \alpha_{11}P_{k-1}$. We will define the two combinations C_k^0 and C_k^1 are independent if and only if the coefficient vectors $(\alpha_{00}, \alpha_{01})$ and $(\alpha_{10}, \alpha_{11})$ are linearly independent. If the two combinations are independent, each node can receive the other combination and create a new combination $\hat{C}_k^i$ in the form $\hat{C}_k^i = \beta_0^i C_k^0 + \beta_1^i C_k^1$, where $i=0$ and $i=1$ represents the two nodes.

Since $\hat{C}_k^i$ is a linear combination of C_k^0 and C_k^1, the new combination $\hat{C}_k^i$ is also a linear combination of packets P_k and P_{k-1}. The new combination vector $[k, \gamma_0^i, \gamma_1^i]$ is saved in the header where $\gamma_0^i = \beta_0^i \alpha_{00} + \beta_1^i \alpha_{10}$ and $\gamma_0^i = \beta_0^i \alpha_{01} + \beta_1^i \alpha_{11}$ $(i=0,1)$. Since the coding coefficients are chosen randomly from a large Galois field, there is high probability that the combinations C_k^i will be different from each other. Each node can create a new combination only once in each epoch. Once a node in G_k creates a new combination, it will not change until it encounters a node in G_j where $j > k$.

6.2. Sequential Recovery Probability

Based on the above model, at any time, the size of group G_k is the same as the number of copies of P_k in the ERP. Hence the probability of the sink encountering one of the nodes in G_k under the NC-ERP is the same as the probability of the sink receiving P_k under the regular ERP, which is equal to the SDP.

Except for the initial packet P_1, the sink node can recover the packet P_k in the NC-ERP either from C_k or C_{k+1}. Recovering P_k from C_k requires the sink to recover P_{k-1} beforehand, or to recover both at the same time by receiving at least two independent combinations C_k. In a similar way, P_k can be recovered from C_{k+1} if P_{k+1} has been recovered beforehand, or P_k can be recovered with the subsequent packets simultaneously.

55

Recovering P_k with the subsequent packets works in a more complicated way. First, P_k can be recovered with P_{k+1} by receiving at least two independent combinations C_{k+1}. Next, if the sink received only one C_{k+1}, P_k can be still recovered by recovering P_{k+1} and P_{k+2} by receiving at last two independent combinations C_{k+2}. This procedure can go on while the sink keeps receiving only one combination for each epoch until the sink receives at least two independent combinations at a certain epoch, say C_{k+N}. Then the sink will be able to recover all N+1 packets from P_k to P_{k+N} simultaneously.

Figure 6.1 depicts the probability of recovering 10 sequential packets while routing 100 packets under the NC-ERP. We recall that the SDP for routing the same number of packets under the regular ERP is 80% and we can see that there is an improvement in packet recovery probability. The recovery probability for the initial packet P_1 is highest, and recovery probabilities for subsequent packets decrease and converge to a certain value, which we refer to as the *Sequential Recovery Probability (SRP)*. Although for the results in Figure 6.1, SRP ($\approx$0.92) is larger than SDP (= 0.8), this may not be true in every case. In the next section, we study the conditions for which SRP > SDP.

6.3.   Deriving Packet Delivery Probability

Based on Figure 5.3(a), suppose the number of copies of P_k reaches a maximum value n at T_{Max} and drops to 0 at T_E. We define q as the probability of the sink encountering a particular node for $0 \le t \le (T_E - T_{Max})$. Since the number of copies increase to n and decrease to 0 close to symmetrical way, we can assume that there are n number of copies in the system for $0 \le t \le (T_E - T_{Max})$, and thus the probability, D, of the sink receiving at least one copy of P_k can be defined as:

$$D = 1 - (1-q)^n. \tag{6.2}$$
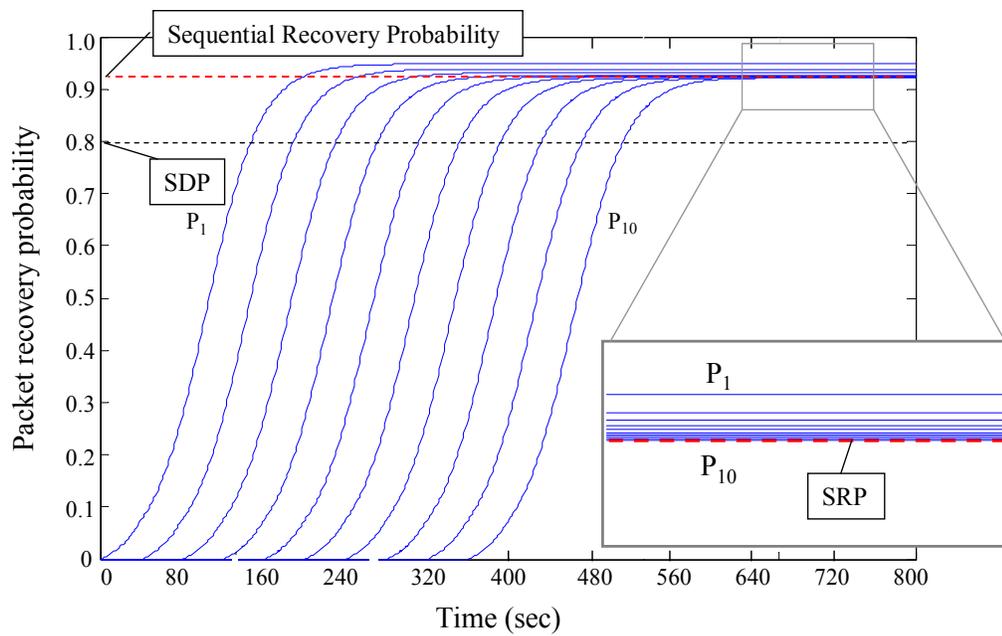
Figure 6.2: Packet Recovery Probability for sequential NC-ERP

Now, if there are n independent combinations C_k instead of n copies of packet P_k, the sink has to receive at least two independent combinations C_k to recover P_k, which will also recover P_{k-1}. The probability of receiving at least two independent combinations is $1 - (1 - q)^n - nq(1 - q)^{n-1}$, which is smaller than the probability D. However, P_k can be also recovered in various ways using combinations C_k and C_{k+1}, as mentioned earlier.

6.4. Deriving Initial Packet Recovery Probability (IRP)

We now calculate the probability of recovering the first packet P_1. Since P_1 is routed without any combination, the probability of the sink receiving P_1 is D; i.e., $\Pr(P_1 \leftarrow P_1) = D$. After the source routes C_2, the sink can also recover P_1 from two independent combinations C_2. At any time, if the sink receives two independent combinations C_k, it can recover all of the packets between P_1 and P_k by receiving a single combination for each epoch from 2 to $(k-1)$; i.e., $C_2, C_3, \ldots C_{k-1}$. Hence, the total probability of recovering P_1 is larger than D.

(a) Initial Recovery Probability

(b) Sequential Recovery Probability
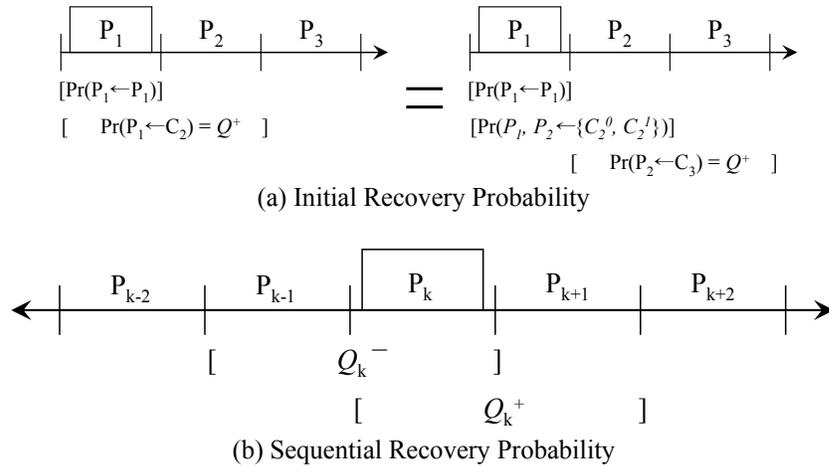
Figure 6.3: Packet Recovery Probability

Let Q^+ be the probability of recovering P_1 from C_2; i.e., $\Pr(P_1 \leftarrow C_2) = Q^+$. As mentioned, the sink can recover P_1 by receiving two independent combinations C_2 with probability $\Pr(P_1,P_2 \leftarrow \{C_2^0, C_2^1\})$, or by receiving only one combination C_2 while recovering P_2 from C_3. As we can see from Figure 6.2(a), recovering P_2 from C_3 is the same as recovering P_1 from C_2, and the probability is Q^+; i.e., $\Pr(P_2 \leftarrow C_3) = Q^+$. Hence, Q^+ can be calculated as follows:

$$Q^+ = 1 - (1-q)^n - nq(1-q)^{n-1} + nq(1-q)^{n-1}Q^+,$$

$$Q^+ = 1 - \frac{(1-q)^n}{1 - nq(1-q)^{n-1}}. \tag{6.3}$$

Next we define Q_1 as the union probability of the sink receiving P_1 and the probability of recovering P_1 from C_2. Since recovering P_1 from C_2 is independent of receiving P_1, Q_1 becomes:

$$Q_1 = D + (1-D) \cdot Q^+ = 1 - \frac{(1-q)^{2n}}{1 - nq(1-q)^{n-1}}. \tag{6.4}$$

Since q and D are in [0, 1], Equation (6.3) shows that Q^+ is in [0, 1], and Equation (6.4) shows that Q_1 is in $[Q^+, 1]$. Equation (6.4) also reveals that $Q_1 \geq D$. In other words, since the SDP $= D$, the probability of receiving P_1 is always improves with the NC-ERP.

Similarly, we can derive Q_2 which is the probability of recovering P_2:

$$Q_2^- = 1 - (1-q)^n - nq(1-q)^{n-1} + nq(1-q)^{n-1}D,$$

$$Q_2 = Q_2^- + (1-Q_2^-) \cdot Q^+, \tag{6.5}$$

where Q_2^- is the probability of recovering P_2 from C_2. P_2 can be recovered from two independent C_2, or by receiving one C_2 together with P_1. Since $0 \leq q \leq 1$ and $0 \leq D \leq 1$, using Equation (6.5) we obtain:

$$Q_2^- = D - nq(1-q)^{n-1}(1-D) = D - nq(1-q)^{2n-1} \leq D.$$
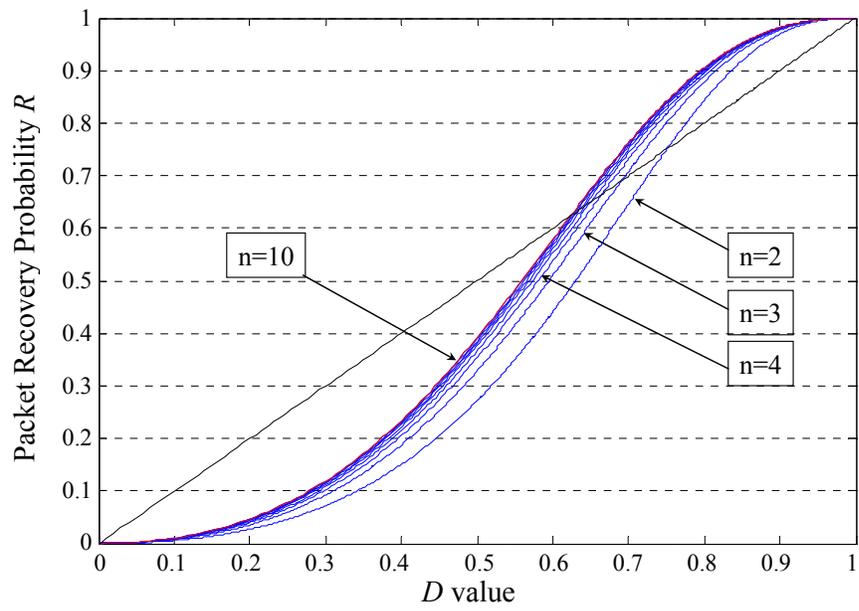
Figure 6.4: Packet Recovery Probability as a function of *D*

Using the inequality $Q_2^- \leq D$, we can derive $Q_2 \leq Q_1$ from Equation (6.4) and Equation (6.5). For $k \geq 3$, Q_k can be derived in a similar way. As depicted in Figure 6.2(b), Q_k is the union of Q_k^- and Q_k^+, where Q_k^- is the probability of recovering P_k from C_k and Q_k^+ is the probability of recovering P_k from C_{k+1}. Q_k^- can be expressed as Q_{k-1}^- which is the probability of recovering P_{k-1} from C_{k-1}, and as already mentioned $Q_k^+ = Q^+$. Hence,

$$Q_k^- = 1 - (1-q)^n - nq(1-q)^{n-1} + nq(1-q)^{n-1} \cdot Q_{k-1}^-,$$

$$Q_k^- - Q_{k-1}^- = nq(1-q)^{n-1} \cdot (Q_{k-1}^- - Q_{k-2}^-),$$

$$Q_k^- = Q_{k-1}^- + \left(nq(1-q)^{n-1}\right)^{k-2} \cdot \left(Q_2^- - D\right).$$

Applying Equation (6.5) to this equation we can derive Q_k by,

$$Q_k = Q_k^- + (1 - Q_k^-) \cdot Q^+. \tag{6.6}$$

Since $Q_2^- \leq D$, we conclude that Q_k^- is a non-increasing function of k. Hence by Equation (6.6), we postulate that Q_k is also a non-increasing function of k, bounded by $[Q^+, 1]$. Thus we have demonstrated that the probability Q_k converges to a certain value, as depicted in Figure 6.1.

6.5. Deriving Sequential Packet Recovery Probability

After routing a large number of packets ($k \gg 1$), the system converges to a state where $Q_k^- = Q_{k-1}^- = Q^-$ and $Q_k^+ = Q_{k+1}^+ = Q^+$, which we define as the steady state. As we can see in Figure 6.2(b), Q^+ and Q^- are two independent identical probabilities in the steady state. Using Q^+ from Equation (6.3) and the steady state convergence $Q^+ = Q^-$, we derive the probability of recovering a P_k in steady state (SRP), R, as follows:

$$R = Q^+ + (1 - Q^+) \cdot Q^- = Q^+(2 - Q^+) = 1 - \left(\frac{(1-q)^n}{1 - nq(1-q)^{n-1}}\right)^2. \tag{6.7}$$

Equation (6.7) can be verified from Equation (6.6) for $k \to \infty$. Using Equation (6.2), we can express Equation (6.7) as a function of D as follows:

$$R(D) = 1 - \left(\frac{1-D}{1+n\left(1-(1-D)^{-1/n}\right)(1-D)} \right)^2 . \tag{6.8}$$

Equation (6.8) is plotted in Figure 6.3 with a different number of combinations n, presenting an S-curve. We can see that the packet recovery probability increases as n increases. The limited value of $R(D)$ as $n \to \infty$ is derived in Appendix B.

6.6. Improvement of SRP

The difference between the SDP and SRP, $I = R - D$, expresses the improvement in packet recovery probability under the NC-ERP compared to the regular ERP. To evaluate the improvement of the NC-ERP, we establish conditions when $I > 0$. Using Equation (6.8), we confirm that the improvement $I = 0$ when $D = 0$ or $D = 1$.

Next, we derive the value of the derivative of I with respect to D when $D = 0$ and $D = 1$. From Equation (6.7) and Equation (6.2), we can derive

$$\frac{dR}{dq} = \frac{2n(1-q)^{2n-1} \cdot \left(1-(1-q)^{n-1}\right)}{\left(1-nq(1-q)^{n-1}\right)^3}, \quad \frac{dD}{dq} = n(1-q)^{n-1},$$

$$\frac{dI(D)}{dD} = \frac{d(R-D)}{dD} = \frac{2(1-q)^n \cdot \left(1-(1-q)^{n-1}\right)}{\left(1-nq(1-q)^{n-1}\right)^3} - 1 . \tag{6.9}$$

From Equatino (6.9), we realize that the derivative of I at $D = 0$ and at $D = 1$ (or, equivalently, for $q=0$ and $q=1$) is -1. Since $I = 0$ at $D = 0$ and at $D = 1$ and I is a continuous function of q, this confirms that for D close to 1, I is strictly positive and for D close to 0, I is strictly negative. Thus, there is a region of D for which the NC-ERP improves packet recovery probability.

Figure 6.4 presents the SRP improvement, I, as a function of D for different numbers of combinations n. We note the two regions of D; the *negative range* where I is negative and the *positive range* where I is positive. We further observe that, as the number of combinations increases, I increases and the positive range of D increases.
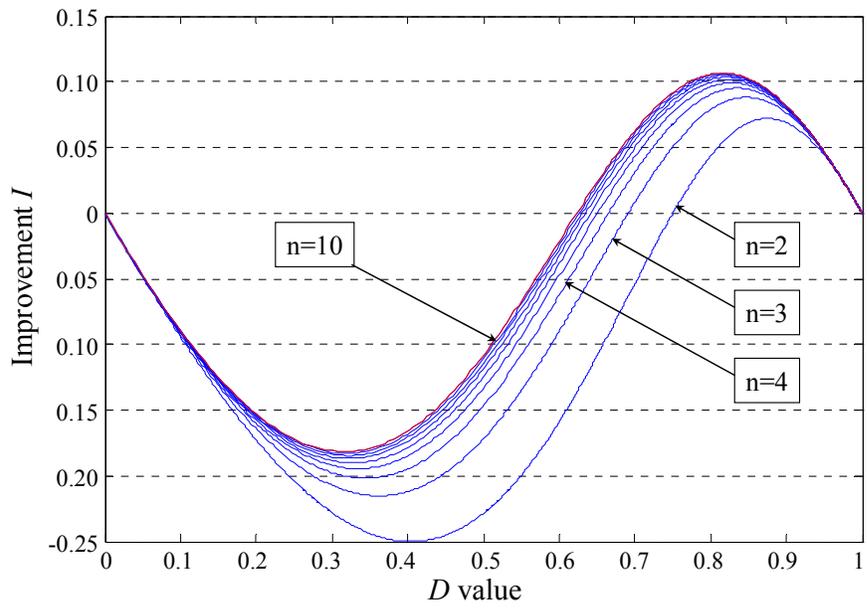
Figure 6.5: Improvement of SRP

6.7.   Packet Recovery Delay

Recovering P_k from C_k can be considered as recovering P_k in the kth epoch. Compared to the regular ERP where P_k cannot be delivered in other than the kth epoch, final recovery of P_k in the NC-ERP can involve a longer delay in terms of epoch if P_k is recovered from C_{k+1}. For example, if the sink does not recover P_k from C_k, and receives one C_{k+1} and two independent C_{k+2}, then P_k is recovered in the $(k+2)^{th}$ epoch which is a delay of two epochs Based on when the sink receives two independent combinations, the delay can be infinitely long.

In the steady state, suppose P_k is recovered in the $(k+e)^{th}$ epoch where e is the delay in epoch. Recall that the probability of receiving only one combination during each epoch is $r_1 = nq(1-q)^{n-1}$, and the probability of receiving at least two independent combinations is $\overline{r_2} = 1-(1-q)^n - nq(1-q)^{n-1}$. The packet recovery is delayed in the e epoch only when P_k is recovered from C_{k+1} by receiving only one combination in each epoch from $(k+1)^{th}$ to $(k+e-1)^{th}$ and receiving at least two independent combinations in the $(k+e)^{th}$ epoch. Hence, the average and the variance of the packet recovery delay in terms of epoch is:

$$E(e) = \frac{(1-Q^-)\overline{r_2}\left(\sum_{e=1}^{\infty} e\, r_1^{e-1}\right)}{Q} = \frac{(1-Q^-)}{Q}\frac{\overline{r_2}}{\left(1-r_1\right)^2}, \tag{6.10}$$

$$E(e^2) = \frac{(1-Q^-)\overline{r_2}\left(\sum_{e=1}^{\infty} e\, r_1^{e-1}\right)}{Q} = \frac{(1-Q^-)}{Q}\frac{\overline{r_2}(1+r_1)}{\left(1-r_1\right)^3},$$

$$V(e) = \frac{(1-Q^-)}{Q}\frac{\overline{r_2}(1+r_1)}{\left(1-r_1\right)^3} - \left(\frac{(1-Q^-)}{Q}\frac{\overline{r_2}}{\left(1-r_1\right)^2}\right)^2. \tag{6.11}$$
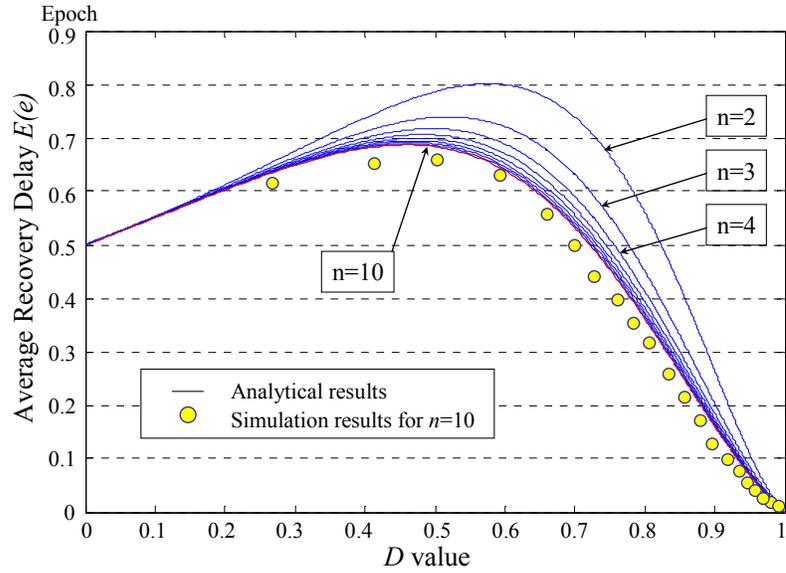
(a) Average Packet Recovery Delay
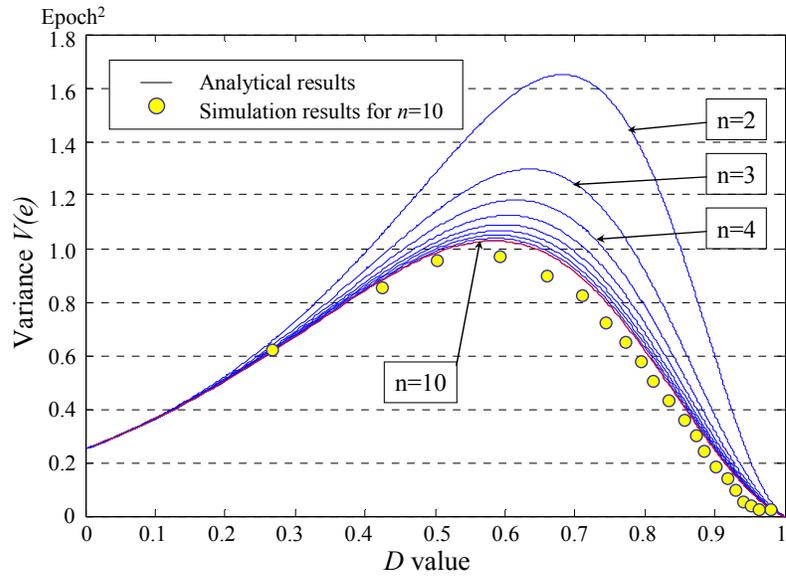
(b) Variance of Packet Recovery Delay

Figure 6.6: Packet Recovery Delay in NC-ERP

65

Figure 6.5 shows the average and the variance of the recovery delay in steady state for different numbers of combinations. We can observe from Figure 6.5(a) that the average delay is zero when $D=1$, and increases as D decreases. An interesting result is that the average delay starts to decrease and converges to $E(e)=0.5$ as $D\rightarrow0$ after some point. Since the probability of receiving a combination decreases as $D\rightarrow0$, the probability of recovering P_k from C_{k+1} with a long delay, $e>>0$, is close to 0. Hence, we can expect that when D is close to 0, P_k is recovered from either C_k or C_{k+1} with $e=1$ only. This fact is also shown in Figure 6.4(b) where $V(e) = 0.25$ and $E(e^2) = 0.5$, which means the probability of P_k being recovered from C_k and the probability of P_k being recovered from C_{k+1} with $e=1$ are equally 50%.

The average delay of packet recovery in the "pre-steady" state is derived in a similar way to deriving Equation (6.10). The only difference from Equation (6.10) is that Q and Q^- is replaced by Q_k and Q_k^- respectively. Since $Q_k^- \geq Q^-$ and $Q_k \geq Q$ we can verify that the average packet recovery delay in the pre-steady state is larger than the average packet recovery delay in the steady state.

6.8.   Combining Three Consecutive Packets

So far we have used the NC-ERP model that combines two consecutive packets to improve packet recovery probability. Next we will use the NC-ERP model that combines three consecutive packets in order to examine the improvement of the packet recovery probability when the number of packets in the combination increases.
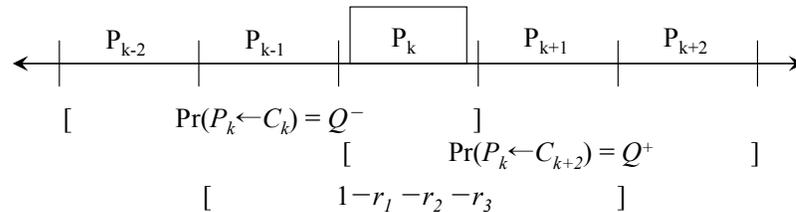
Figure 6.1: Packet Recovery for 3 packet combinations

Figure 6.6 shows how to derive the steady state packet recovery probability of the NC-ERP that combines three consecutive packets. Similar to NC-ERP that combines two consecutive packets, the probability of recovering P_k, is union of the probabilities of recovering P_k from C_k, C_{k+1}, and C_{k+2}. The probability of recovering P_k from C_k, Q^-, and the probability of recovering P_k from C_{k+2}, Q^+, are independent. However, the probability of recovering P_k from C_{k+1} is not independent of the other two probabilities. Instead of deriving the probability of recovering P_k from C_{k+1}, we only need to derive the probability of recovering P_k by receiving three independent C_{k+1}.

Recall the probability of not receiving any combination in an epoch $r_0 = (1-q)^n$, the probability of receiving only one combination in an epoch $r_1 = nq(1-q)^{n-1}$, and the probability of receiving two combinations in an epoch $r_2 = {}_nC_2 \cdot q^2(1-q)^{n-2}$. In the steady state, Q^- and Q^+ are two independent identical probabilities which are:

$$Q^+ = (1 - r_0 - r_1 - r_2) + (r_1 + r_2)Q^+ + r_2 r_0 Q^+$$

$$Q^+ = Q^- = \frac{1 - r_0 - r_1 - r_2}{1 - r_1 - r_2 - r_2 r_0} = 1 - \frac{r_0(1 - r_2)}{1 - r_1 - r_2 - r_2 r_0}.$$

In order to recover P_k not depending on C_k or C_{k+2}, the sink has to receive at least three independent combinations C_{k+1}, and the probability of this event is $1 - r_0 - r_1 - r_2$. Hence, the SRP of the NC-ERP combining three packets, R_3, is:

$$R_3 = Q^+ + Q^- - Q^+ Q^- + (1 - Q^+)(1 - Q^-)(1 - r_0 - r_1 - r_2),$$

$$R_3 = 1 - \frac{r_0^2(1 - r_2)^2}{(1 - r_1 - r_2 - r_2 r_0)^2}(r_0 + r_1 + r_2). \tag{6.12}$$

Equation (6.12) is plotted in Figure 6.7 as a function of D, which is the S-curve on the right compared with Equation (6.8) which is the S-curve on the left. From this observation, we can expect that as the number of packets in the combination of the NC-ERP increases, the packet recovery probability $R(D)$ will have a steeper slope where the slope gets closer to $D=1$.
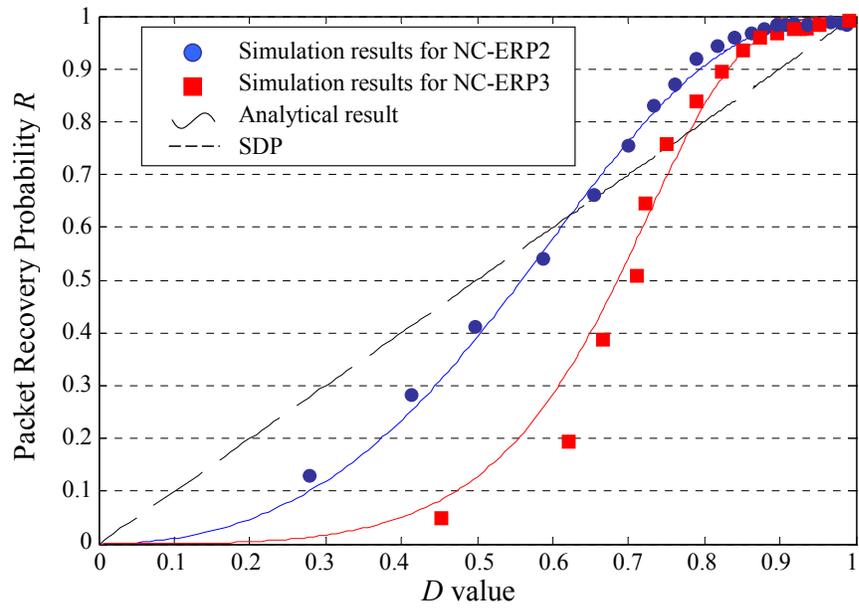
Figure 6.7: Comparison of Sequential Recovery Probability for $n = 10$

6.9.   Results

The simulations were done in our random mobility model. The network area is a 1000m x 1000m closed torus-like square and the network consists of N=50 mobile nodes plus one sink. Each node has a transmission range of 25m and is able to store only one packet in its memory. The direction of each node is uniformly distributed in $[0, 2\pi]$ and the speed is uniformly distributed in $[20, 50]$m/s. Each node changes its velocity with a rate of 0.2/s. In this random mobility model, the encounter rate λ between two particular nodes is 0.001/sec. The Galois field is a set of integers from 1 to 1000. All the simulation results are averaged over 1000 trials.

Figure 6.8 shows the simulation results of recovery probabilities for the first 15 packets under the NC-ERP, compared with the analytical results which we derived from Equations (8)−(10). Both analytical and simulation results show that the initial packet recovery is the highest, and then it decreases for the subsequent packets and converges to the SRP. Figure 6.8(a) depicts a 6% improvement of the NC-ERP when the SDP value is 70%. However, when the SDP value is 50%, as stated in Figure 6.8(b), we can see that the improvement of the NC-ERP is −0.9%. Both analytical and simulation results show that the packet recovery probability converges with the SRP

Two different simulation results of the packet recovery probabilities in steady state, SRP, are plotted in Figure 6.7. The solid S-curves indicate our analytical results, and the small dots indicate the simulation results where the circle, NC-ERP2, is the SRP of the NC-ERP that combines two packets and the square, NC-ERP3, is the SRP of NC-ERP that combines three packets. Both simulation results are close to the analytical result, especially for D close to 1, showing a positive range and a negative range of improvement, SRP-SDP. This verifies that our analytical model and solutions are accurate.
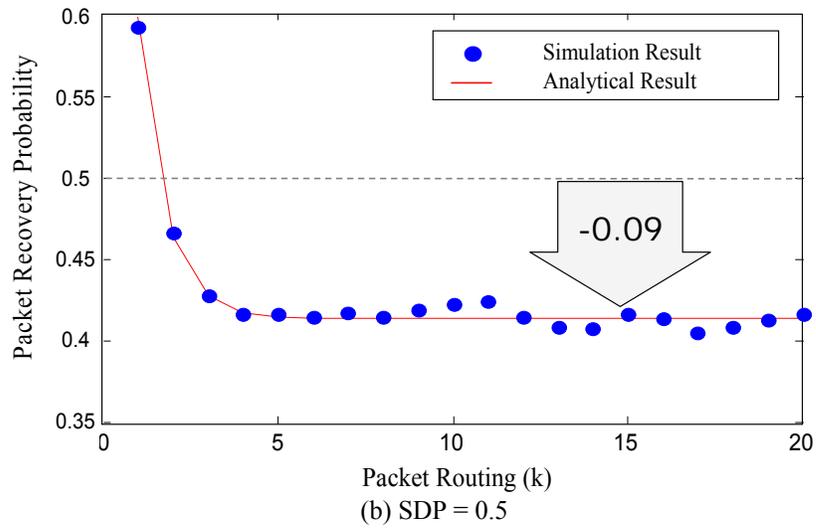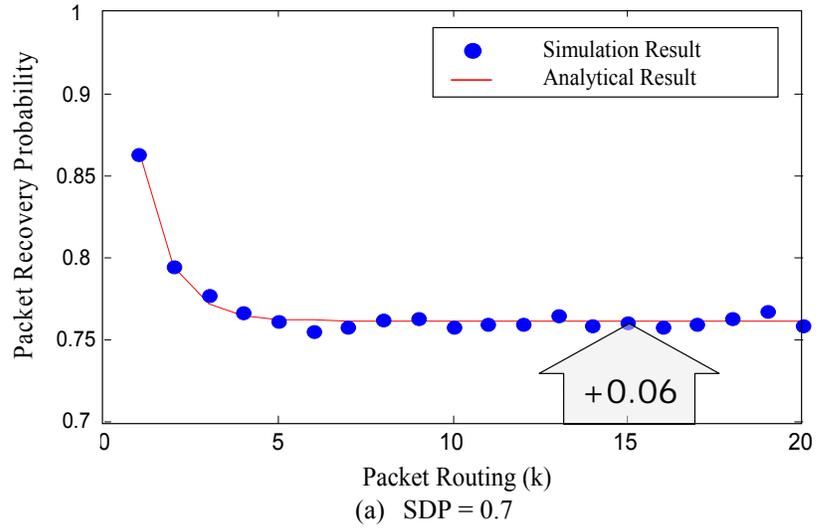
Figure 6.8: Initial Recovery Probability for *n* = 10

Comparing the positive range and the improvement of the SRP−SDP, both
simulation and analytical results show that the NC-ERP that combines two packets is
more effective than the NC-ERP that combines three packets.

Simulation results for the average and variance of packet recovery delay are
plotted in Figure 6.6. The simulation results show that the analytical result derived
from Equation (6.10) and (6.11) are quite accurate, however, both the average and
variance of packet recovery delay for n=10 is slightly smaller than the analytical
results. In our random mobility model simulation, when the source encounters the sink,
instead of transmitting a combination C_k, the source can transmit both P_k and P_{k-1} to
the sink, which results in 0 delay for P_k. This reduces the average and variance of
packet recovery delay and makes a gap between the simulation and analytical results.

Increased packet recovery delay is a disadvantage of the NC-ERP; however, the
average delay is less than one epoch and the standard deviation is less than 1.3 epochs.
For a DTN using the NC-ERP, we can assume that this delay is tolerable.

CHAPTER 7

Related Works

In the last few years there have been many studies on Mobile Sensor Networks (MSNs). MSNs, which are often referred to as spin off of Mobile Ad Hoc Networks (MANET), consist of mobile nodes which communicate with peer nodes when they come into each others' communication range. Data packets in MSNs are routed from source nodes to sink nodes by being relayed through the peer mobile nodes. However, due to the mobility of the nodes, the topology of a mobile network frequently changes and routing data packets in MSNs and in MANETs becomes a challenging problem. When two communicating nodes move out of each others' transmission range, the link between the two nodes becomes disconnected and the end-to-end path breaks. For MANET, OLSR [59,60], AODV [61], DSR [62], ZRP [63], and numerous other protocols [64-67] have been proposed as solutions to this challenge.

However, these protocols assume that the network remains connected, so that an end-to-end routing path exists between the source and the destination (the sink) nodes. In a sparse network, for example, when the transmission range of the nodes is short, the number of neighbors of a node is small to the extent that connections between nodes become intermittent. Consequently, packets are transmitted from one node to another relying on the mobility of the nodes and on the contingency of encountering other nodes. Such a networking paradigm is often referred to as "store-carry-forward," [2, 15] as opposed to the traditional "store-and-forward" paradigm. Unfortunately, the "store-carry-forward" routing results in an increased end-to-end packet delivery delay, which is appropriate only for particular set of applications. A Delay Tolerant Network

(DTN) is an intermittently connected network that can tolerate such increased and unpredictable end-to-end delay.

Several studies have addressed the properties and advantages of the DTN [40-43, 68,69]. Routing protocols for DTN focus mostly either on finding the shortest end-to-end routing path in DTN, or on reducing the total energy consumed for packet transmissions at the expense of an increased end-to-end delay.

If the behavior of the mobile nodes is determined or predictable, future transmissions between nodes can be scheduled ahead of time using deterministic routing protocols [44-47]. Assuming that future network characteristics can be known, [44] proposes six types of routing algorithms, depending on the amount of knowledge of the network. This knowledge contains information about past contacts between nodes, their queuing occupancies, and future traffic demands. Since network characteristic cannot be predicted over infinitely long time, [45] proposed a routing algorithm that finds the best path by looking ahead over fixed time interval T. These deterministic routing algorithms select the end-to-end path using dynamic programming and a shortest path algorithm before the source transmits its data packet.

In most cases, behavior of mobile nodes is random and unpredictable, and the network requires a stochastic routing protocol in which a relaying node dynamically decides on its next recipient. A relaying node can choose its recipient based on mobility pattern, encounter history, or other information. Algorithms that were proposed in [48-52] use one-hop information, while [53-55] accumulate end-to-end information. However, when node mobility is totally random, it may be difficult to choose one possible recipient over another. Additionally, use of special nodes with high mobility and capacity, which allows more reliable communication than use of peer nodes, has been advocated by some researchers [38,39,46,47,56-58].

The Epidemic Routing Protocol (ERP) [6] is a DTN protocol which shows the shortest end-to-end delay in a totally random mobile network. Nevertheless, ERP results in increased expense of resources such as energy consumption of packet transmissions and network capacity. To overcome this drawback of ERP, *SWIM* [2, 15] uses a small sized anti-packet to restrict packet replication. In the *Spray and Wait* routing protocol [17], packet copies are propagated only during the Spray period, while in the Wait period nodes can communication only with the sink node.

Controlling packet flow in mobile networks can also restrict packet replication in ERP [6, 19-22]. When nodes are equipped with limited amount of energy, controlling packet flow based on nodes' energy information improves the efficiency of packet routing [34]. Similarly, controlling packet flow based on nodes' residual battery information can increase network lifetime [21].

Coding-based protocols have been applied to wireless networks in various ways. Network coding increases the network throughput [23-28] and also it can compress multiple packets in the limited node memory [29-32]. Network-coding applied protocols create multiple independent combinations to reduce the total number of transmissions, i.e. increase the throughput. The size of a combination is the same as that of the original packet, but the total number of transmissions can be reduced, when there are multiple data packets being routed simultaneously to multiple sinks [29-31]. Erasure-coding [33] uses smaller sized packet "fragments" which carries partial information of the original packet. Since a data packet creates multiple fragments, this may increase the total number of transmission. In any case, the sink has to receive more than one packet to recover an original data packet. Both erasure-coding and network-coding protocols require the sink to receive more than one packet to recover an original data packet.

CHAPTER 7

Summary, Conclusions and Future Research

Based on the memoryless property of encounter times between nodes in an intermittently connected network, a Markov Chain model was used to analyze the Epidemic Routing and point out the drawbacks of the (unrestricted) Epidemic Routing scheme. Although ERP shows high packet delivery probability, it consumes excessive amount of node energy.

In this study, a number of schemes were proposed to eliminate the redundant packet replicas in ERP in order to reduce the energy consumption by the network nodes and, thus, increase the lifetime of the network. In general, there is a fundamental tradeoff among the three performance parameters: the energy (as expressed by the number of packet copies), the packet delivery delay, and the packet delivery probability.

In order to find the most efficient way to tradeoff between the number of copies and the packet delivery delay for fixed packet delivery probability, three different methods were designed to restrict the ERP in generating copies – the Restricted Epidemic Routing (RER) schemes. For each RER scheme (EX, LT, and LC scheme) a matched Markov Chain model was used to calculate the average number of copies in the system and the packet delivery probability at a given time. Both the simulation and the analytical results indicated that the LC-scheme, which explicitly limits the number of copies to some pre-determined number, is the scheme with the most efficient tradeoff function between the number of copies (i.e., energy expenditure) and the packet delivery delay.

The discussion considered a single packet routing with sufficiently large amount of battery energy and storage. In general, the amount of battery energy and storage are limited, and the network continually consumes energy and storage while routing multiple packets. This leads to the deterioration in network performance. In order to compare the performance of the various schemes, two thresholds were defined, the Target Delivery Probability (TDP) and the Minimum Delivery Probability (MDP).

Comparing the TDP with the packet delivery probability of sequential ERP, one can determine whether the network can conserve more energy or not. If the packet delivery probability exceeds the TDP, then the energy consumption can be reduced at the expense of reducing the packet delivery probability to the TDP. Otherwise, the network requires extra process instead, in order to increase the packet delivery probability.

Even when the packet delivery probability exceeds the TDP, with time, nodes will have their batteries depleted and become inactive. Consequently, the packet delivery probability will gradually decrease, crossing the MDP level at some point in time. This time was defined as the MDP lifetime, and the length of the MDP lifetime was used to evaluate the energy efficiency of the RER schemes. The simulation results showed that although the RER schemes can extend the lifetime of ERP, the gap between the MDP lifetime and the *ideal lifetime* increased.

Owing to the observation that *ideal lifetime* can be achieved when all the nodes are depleted of energy at approximately the same time, using the residual battery energy information showed further increase of the MDP lifetime. This result suggests that the lifetime of a network could be extended if the variance of the number of transmissions for each node is decreased. Combining the residual battery energy information of the nodes is one method that can decrease this variance and, hence, extend the lifetime of the network.

76

In distinction from the previous case, when the packet delivery probability does not exceed the TDP, typically it is impossible to increase the packet delivery probability to the TDP using RER schemes. When the nodes have limited storage, nodes have to remove old packets in order to replicate new packets. If the packet generation rate increases over a certain threshold, the packet may not have enough time to be delivered at the sink before it gets removed permanently from the network. Consequently, the packet delivery probability decreases below the TDP, even when the nodes are all active.

In order to improve the ERP of which packet delivery probability is below the TDP, the *Linear Network Coding* can be applied to ERP (LNC-ERP). Based on the analysis of the LNC-ERP, the *Initial Delivery Probability* and the *Sequential Recovery Probability* can be derived. These probabilities showed that there is a range of packet delivery probability where LNC-ERP can improve on the *Sequential Delivery Probability* of basic ERP. However, combining more than two packets showed less improvement than combining two packets.

Summing up, when the packet delivery probability exceeds the TDP, schemes that focus on minimizing energy consumption at individual nodes may not be as efficient at extending the network lifetime as schemes that tend to equalize the energy expenditure at the network nodes. The latter types of schemes are of particular importance in case where the node "mixing" is limited and some nodes are likely to be depleted of their energy much sooner than other nodes. When the packet delivery probability is below the TDP, linear network coding can improve the packet delivery probability. This advantage of applying linear network coding to ERP can be more general. As long as the network sequentially generates multiple packets and the sink receives more than one of these packets with high probability, linear network coding can improve the packet delivery probability.

This research can be extended in the following directions.

- **Other ways of splitting the number of copies in LC-scheme.**

  Recall that the LE-scheme is a modified LC-scheme that splits the number of copies in proportion to residual energies of the nodes. Although LE-scheme is more efficient than the other RER schemes, it may not be the optimum scheme. The number of copies can be split based on other ways such as the current encounter rate, current velocity, or estimated distance from the sink.

- **Multiple sources and larger memory size.**

  The LNC-ERP proposed in this work was based on the assumption that there is only one source node and the relaying nodes can store only one packet in their memory. If the number of source nodes increases, the number of possible combinations in an epoch increases and requires more memory.

- **Random packet generation.**

  In the sequential ERP, the packet generation rate was assumed to be fixed, either high or low. If the source generates and starts to route at a random time, the time interval between each packet generation, T_d, becomes a random value, and the packet delivery probability may not converge to a fixed limit. Based on a certain threshold of T_d, the source may combine the current packet with the previous packet only if T_d is shorter than the threshold.

- **Security Issue.**

  In ERP, it is possible for a malicious node to intercept the packet when the node carrying a packet comes into contact. As the number of copies increases, the intercepting probability increases as well. Assuming the intercepting probability is less the packet delivery probability at the sink, the study of LNC-ERP shows that it is possible to reduce only the intercepting probability.

APPENDIX A

Memoryless Property of Encounter Rate

A.1. Encounter rate between two nodes

When a node is ready to transmit a packet, the node will wait for some random time (encounter time) until the node comes within the transmission range of another node. Groenevelt *et al.* [70] have shown that the inter encounter time between any pair of nodes can be assumed to be exponentially distributed if the following three conditions hold. First, nodes move according to a random mobility model. Second, node transmission range is small compared to the area of the network. Third, the speed of nodes is sufficiently high. Next, using simulation, we show that this assumption is reasonable.

In our simulations, time is quantized into discrete time slots where each time slot is a 1 [sec] interval. The total network area is a 1000 [m] x 1000 [m] closed square-shape area, where each side is connected to the opposite side creating a torus-like structure. At time t=0 [sec], we randomly place two nodes, each of which with transmission range of 25 [m]. In our random mobility model, every second every node changes its velocity independently with the probability of 20%, so that the average duration between velocity changes is 5 [sec]. Each node chooses its direction uniformly from [0,2π] and its speed uniformly from [20,70] [m/s]. Thus the average speed of the nodes is 45 [m/s].

Fig. 19 depicts the simulation result of the probability distribution of encounter times between two nodes and compares it with the exponential distribution. The simulation result, which is plotted as gray dots, is based on one million observed outcomes of encounter times, $T_E > 0$. The black solid line is the exponential

distribution with parameter λ, which is calculated by the inverse of the average encounter time, $\lambda = 1/E[T_E \mid T_E > 0] = 0.00127/\text{sec}$. Assuming that expected values are exponentially distributed, when we divide the time axis into 100 intervals, where each interval has a length of 10 sec, the chi-square statistic is 33.47. The chi-square test threshold for 5% significance with degrees of freedom 99 is 123.23. Since the chi-square statistic does not exceed the threshold of 33.47, we conclude that random variable T_E follows exponential distribution with very high probability.

A.2. Encounter time between two groups of nodes

In a mobile network, there are usually more than two nodes. Since nodes move randomly, independently, and homogeneously, we can assume that the encounter rate between any two nodes has the same exponential distribution. We define as encounter of two groups of nodes, as an event where a node from each of the two groups come in close enough proximity to communicate. Hence, the encounter rate between two different groups of nodes will be proportional to the number of nodes in each group. As with the encounter between two nodes, we will show that this assumption is reasonable through simulation results.

Suppose that there are *N* nodes in the system. Then, the random variable of the next encounter time between the *N-1* nodes and one particular node (the source node) can be thought of as the minimal time of *N-1* independent random variables, each one denoting the next encounter time between a node from the *N-1* node group and the source node. Since the encounter time between any two nodes follows the exponential distribution, the *cdf* of the encounter times between the source node and the *N-1* peer nodes ($T_{1,N-1}$) can be derived by using the *cdf* of the encounter time between two nodes ($T_{1,1}$); i.e.,

$$\Pr[T_{1,N-1} \geq t] = \Pr[T_{1,1} \geq t] \wedge (N-1) = (e^{-\lambda t})^{N-1} = e^{-(N-1)\lambda t}$$

Hence, the pdf of the encounter time between source and the other peer nodes also follows exponential distribution with parameter $(N-1)\lambda$, where λ is the encounter rate between any two nodes. In a similar way, we can also derive the *cdf* of the encounter time between two different groups of nodes. Suppose that the number of nodes in one group is k and that the number of nodes in the other group is $N-k$. Then, the encounter time between these two groups $(T_{k,N-k})$ can be calculated as:

$$\Pr[T_{k,N-k} \geq t] = \Pr[T_{1,N-k} \geq t]^{\wedge} k = (e^{-(N-k)\lambda t})^k = e^{-k(N-k)\lambda t}$$

Suppose there are a total of N nodes in the network, where group A consists of k number of nodes and group B consists of $(N-k)$ nodes. Then the *pdf* of the encounter time between the two groups of nodes follows exponential distribution with parameter $k(N-k)\lambda$, where λ is the encounter rate between two nodes. This is demonstrated by the following simulation results in which there are N=50 mobile nodes and the encounter rate is $\lambda = 0.00127$/sec. Simulation results, plotted in Fig. 20 as dotted gray line, are compared with the exponential distribution with $\lambda = 0.00127$/sec. The simulation results confirm our analytical derivation. The gap between the simulation result and the exponential distribution is due to the overlapping between transmission areas of the two groups, A and B. The encounter rate between the two groups, which contain k and $(N-k)$ nodes, respectively, is proportional to $k(N-k)$. When k=1 or k=49, there is maximal overlap within one of the two groups, while the overlap at both group is minimal when the number of nodes in the two groups is equal, so that k=$(N-k)$=25.
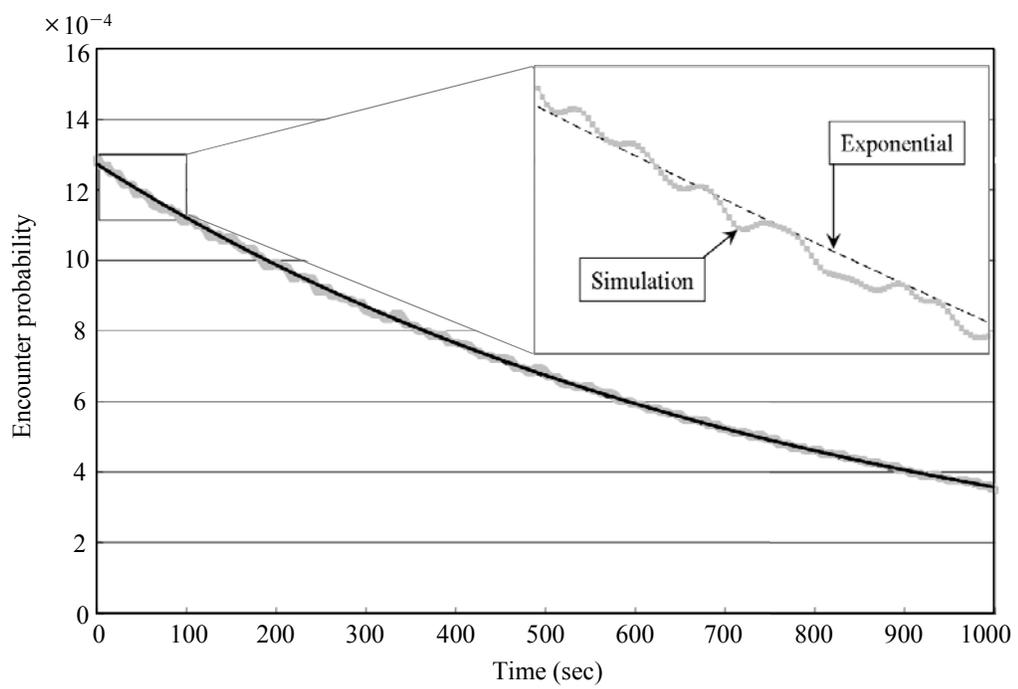
Figure A.1: Probability distribution of encounter time between two nodes
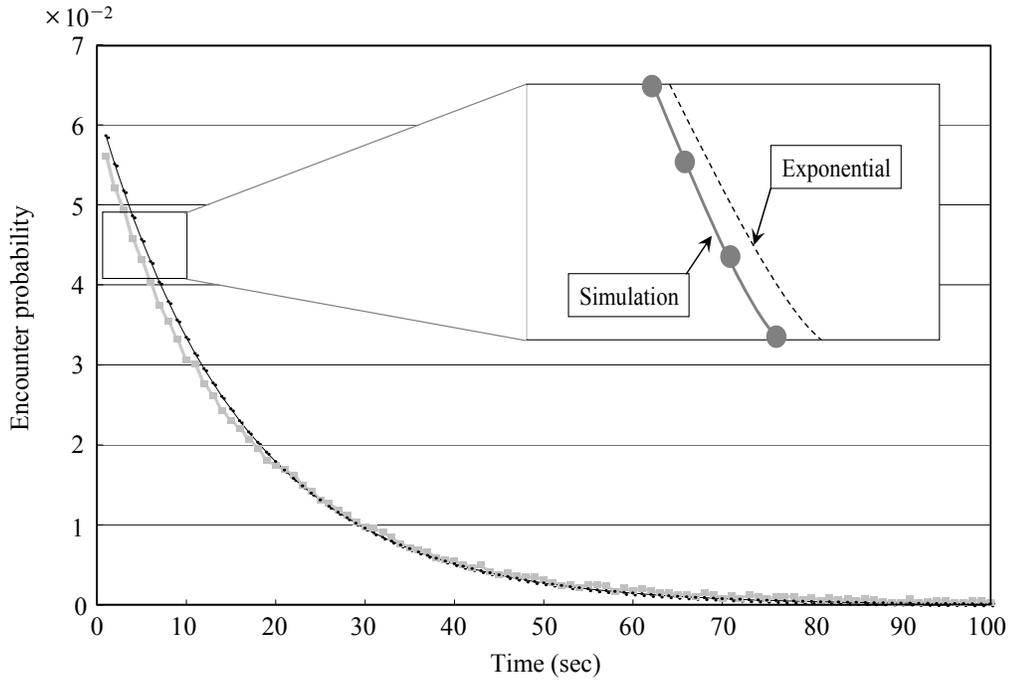
(a) k = 1, (N-k) = 49
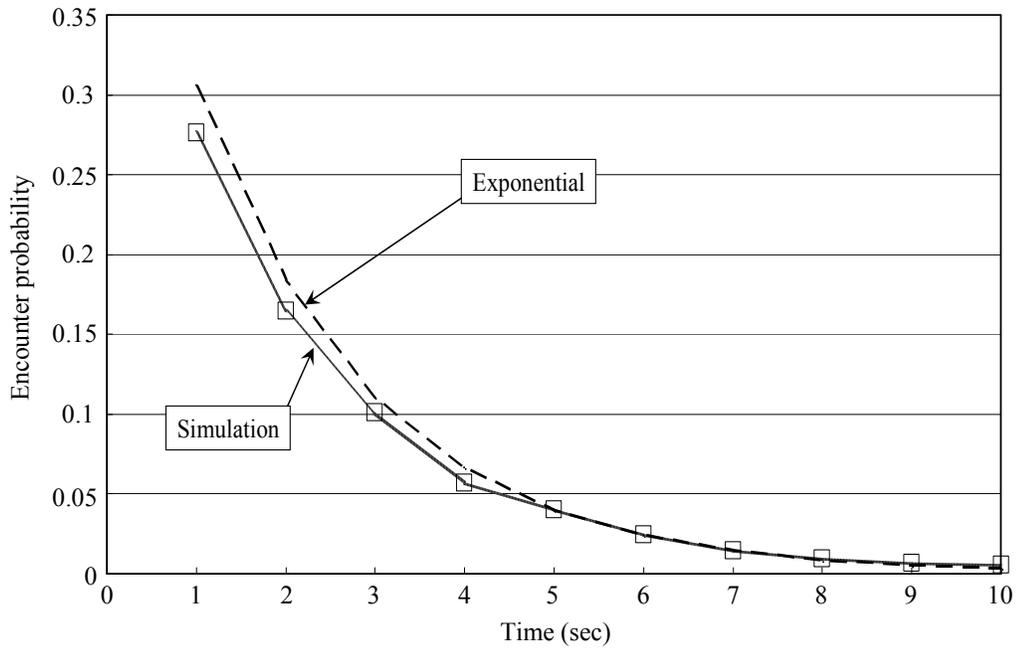
(b) k = 10, (N-k) = 40

Figure A.2: Probability distribution of encounter time between two groups of nodes

A.3. Encounter rate in other random mobility models

Simulation results for other random mobility modes show that memoryless property depends mostly on the randomness of the model. For example, for the Gauss-Markov random mobility model [71], the chi-square test result differs according to the tuning parameter of randomness α. When α=1, the velocity of the node is fixed, while when α=0, the velocity of the nodes changes with each adjustment in a totally random manner. The λ value that corresponds to α=0.2 is 0.00124/sec, and the λ value that corresponds to α=0.8 is 0.00107/sec. Comparing the encounter times with the exponential distribution, when we divide the time axis into 100 intervals, the chi-square statistic for α=0.2 is 80.15, which does not exceed the chi-square test threshold for 5% significance of 123.23. However, when α increases to 0.8, the chi-square statistic becomes 647.56, which exceeds the value of the threshold of 123.23. Based on the chi-square test, the random variable T_E follows the exponential distribution for α=0.2, but not for α=0.8. We conclude that the probability distribution of encounter times becomes closer to the exponential distribution when the movement of the node tends to resemble more the random mobility.

APPENDIX B

Limited Value of Packet Recovery Probability and Improvement

Recall Equation(12), which is the SRP as a function of the SDP,

$$R(D) = 1 - \left(\frac{1-D}{1 + n\left(1 - (1-D)^{-\frac{1}{n}}\right)(1-D)} \right)^2 .$$

In order to derive the converging function of $R(D)$, we first derive the limiting value of $n(1 - (1-D)^{-1/n})$. By transposing $z = (1-D)^{-\frac{1}{n}}$, we can derive that $n = -\left(\log_{(1-D)} z\right)^{-1}$ and n→∞ is equivalent to z→$^+$1. Based on the l'Hôpital's rule,

$$\lim_{n \to \infty} n\left(1 - (1-D)^{-\frac{1}{n}}\right) = \lim_{z \to {}^+1} \frac{z-1}{\log_{1-D} z} = \lim_{z \to {}^+1} \frac{1}{z^{-1} \log_{1-D} e} = \ln(1-D) .$$

Hence, Equation (5.8) converges to:

$$\lim_{n \to \infty} R(D) = 1 - \left(\frac{1-D}{1 + (1-D)\ln(1-D)} \right)^2 .$$

Based on this, the improvement I converges to:

$$\lim_{n \to \infty} I(D) = (1-D) - \left(\frac{1-D}{1 + (1-D)\ln(1-D)} \right)^2 .$$

The converging function of $R(D)$ is is plotted in Figure B.1, and we can see that even for large number of n, there always exists a negative range of D, where the D value is close to 0.
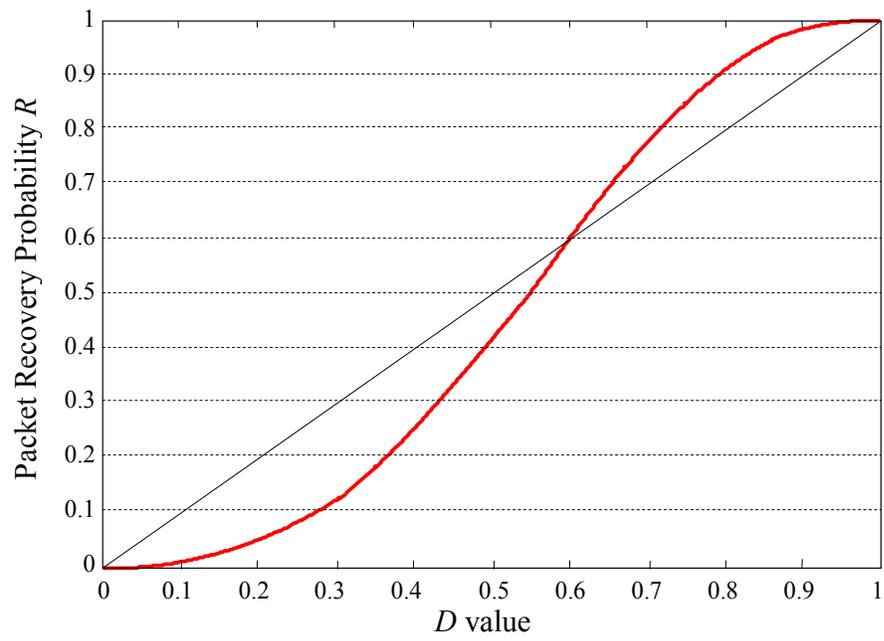
Figure B.1: Sequential Recovery Probability for n→∞

BIBLIOAGRAPHY

[1]   P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rebenstein, "Energy Efficient Computing for Wildlife Tracking: Design tradeoffs and Early Experiences with ZebraNet," ASPLOS-X, San Jose, CA, Oct. 2002.

[2]   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)," MobiHoc, Annapolis, MD, June 1-3, 2003.

[3]   S. M. Nazrul Alam and Zygmunt J. Haas, "Coverage and Connectivity in Three-Dimensional Underwater Sensor Networks", Wireless Communication and Mobile Computing (WCMC) Journal, Volume 8 Issue 8, Oct. 2008.

[4]   S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay Tolerant Networking: An Approach to Interplanetary Internet," IEEE Communications Magazine, vol. 41, 2003.

[5]   A. Pentland, R. Fletcher, and A. Hasson, "A Road to Universal Broadband connectivity," 2nd International Conference of Open Collaborative Design for Sustainable Innovation; Development by Design, Dec. 2002.

[6]   A. Vahdat and D. Becker, "Epidemic Routing for Partially Connected Ad Hoc Networks," Technical Report, Duke University, Apr. 2000.

[7]   D. J. Daley and J. Gani, "Epidemic Modeling," Cambridge University Press, 1999.

[8]   M. E. J. Newman, "Spread of Epidemic Disease on Networks," Physical Review E 66, 016128, 2002.

[9]   L. A. Meyers, "Contact Network Epidemiology: Bond Percolation Applied to Infectious Disease Prediction and Control," Bulletin of the American Mathematical Society, vol. 44, Jan. 2007.

[10]   M. E. J. Newman, I. Jensen, and R. M. Ziff, "Percolation and epidemics in a two-dimensional small world," Physical Review E 65 021904, 2002.

[11]   A. Jindal and K. Psounis, "Performance Analysis of Epidemic Routing under Contention," IEEE Workshop on Delay Tolerant Mobile Networks, Vancouver, Canada, July 2006.

[12]   D. Nain, N. Petigara, and H. Balakrishnan, "Integrated Routing and Storage for Messaging Applications in Mobile Ad Hoc Networks," WiOpt, Autiplis, France, Mar. 2003.

[13]   J. Su, A. Chin, A. Popivanova, A. Goel, and E. de Lara, "User Mobility for Opportunistic Ad-Hoc Networking," 6th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), English Lake District, UK, Dec. 2-3, 2004.

[14]   T. Small and Z. J. Haas, "Resource and Performance Tradeoffs," ACM SIGCOMM, Philadelphia, Pennsylvania, Aug. 22-26, 2005.

[15]   Z. J. Haas and T. Small, "A New Networking Model for Biological Applications of Ad Hoc Sensor Networks," IEEE/ACM Transactions on Networking, vol. 14, no. 1, Feb. 2006.

[16]   S. K. Yoon and Z. Haas, "Efficient Tradeoff of Restricted Epidemic Routing in Mobile Ad Hoc Networks," Milcom, Orlando, FL, Oct. 2007.

[17]   T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks," ACM SIGCOMM workshop on Delay Tolerant Networking, Philadelphia, PA, Aug. 22-26, 2005.

[18]   T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility," PerCom Workshop on Intermittently Connected Mobile Ad Hoc Networks (ICMAN), White Plains, NY, Mar. 2007.

[19]   K. A. Harras, K. C. Almeroth, and E. M. Belding-Royer, "Delay Tolerant Mobile Networks (DTMNs): Controlled Flooding in Sparse Mobile Networks," IFIP-TC6 Networking Conference, vol. 3462, pp. 1180–1192, Waterloo, Canada, May 2-6, 2005.

[20]   Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based Ad Hoc Routing," IEEE INFOCOM, New York, NY, June 23, 2002.

[21]   S. K. Yoon, Z. Haas, and Jae H. Kim, "Tradeoff between Energy Consumption and Lifetime in Delay Tolerant Mobile Network," Milcom, San Diego, CA, Nov. 2008.

[22]   J. Leguay, T. Friedman, and V.Conan, "DTN Routing in a Mobility Pattern Space," in WDTN '05: SIGCOMM'05 DTN workshop, pp. 276-283, 2005.

[23]   J. S. Park, D. S. Lun, Y. Yi, M. Gerla, and M. Médard, "CodeCast: A network Coding Based Ad Hoc Multicast Protocol," Wireless Communications, IEEE, vol. 13, no. 5, pp. 76–81, October 2006.

[24]   P. A. Chou and Y. Wu. "Network Coding for the Internet and Wireless Networks", MSR-TR-2007-70, Microsoft Research, Jun. 2007.

[25] S. Deb, M. Effros, T. Ho, D. R. Karger, R. Koetter, D. S. Lun, M. Médard, and N. Ratnakar, "Network Coding for Wireless Applications: A Brief Tutorial," in IWWAN, 2005.

[26] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A Random Linear Network Coding Approach to Multicast," IEEE Transactions on Information Theory, vol. 52, no. 10, pp. 4413–4430, October 2006.

[27] X. Li, T. Jiang, Q. Zhang, and L Wang, "Binary Linear Multicast Network Coding on Acyclic Networks: Principles and Applications in Wireless Communication Networks," IEEE Journal on Selected Area in Communications, IEEE Journal on Selected Areas in Communications, vol. 27, no. 5, pp. 738-748, Jun. 20 2009.

[28] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the Air: Practical Wireless Network Coding," Proceedings of the ACM SIGCOMM Conference, pages 243–254, Pisa, Italy, August 2006.

[29] Y. Lin, B. Liang, and B. Li, "Performance Modeling of Network Coding in Epidemic Routing," Proc. 1st International MobiSys Workshop on Mobile Opportunistic Networking, San Juan, Puerto Rico, June 11, 2007.

[30] Y. Lin, B. Li, and B. Liang, "Stochastic Analysis of Network Coding in Epidemic Routing," IEEE Journal on Selected Area in Communications, Special Issue on Delay and Disruption Tolerant Wireless Communication, vol. 26, no. 5, pp.794–808, June 2008.

[31] S. Ahmed, S. S. Kanhere, "HUBCODE: message forwarding using hub-based network coding in delay tolerant networks," Proceedings of the 12th ACM International conference on Modeling Analysis and Simulation of Wireless and Mobile Systems, Canary Islands, Spain, 2009.

[32] S. K. Yoon and Z. J. Haas, "Application of Linear Network Coding in Delay Tolerant Networks," Invited at the 2nd International Conference on Ubiquitous and Future Networks, Jeju Island, Korea, June 16-18, 2010.

[33] Y. Wang, S. Jain, M. Martonosi, and K. Fall, "Erasure-Coding Based for Opportunistic Networks," Proc. ACM SIGCOMM, Workshop on Delay Tolerant Networking and Related Topics (WDTN-05), Aug. 2005.

[34] J. Chang and L. Tassiulas, "Energy Conserving Routing in Wireless Ad-hoc Networks," INFOCOM, Conference on Computer Communications, Tel Aviv, Israel, Mar. 2000.

[35]   R. C. Shah and J. M. Rabaey, " Energy Aware Routing for Low Energy Ad Hoc Sensor Networks," IEEE Wireless Communications and Networking Conference, Orlando, FL, Mar. 17-21, 2002.

[36]   M. Maleki, K. Dantu, and M. Pedram, "Power-aware Source Routing Protocol for Mobile Ad Hoc Networks," International Symposium on Low Power Electronics and Design, Monterey, CA, Aug. 12 - 14, 2002.

[37]   S. Jain, M. Demmer, R. Patra, and K. Fall, "Using Redundancy to Cope with Failures in a Delay Tolerant Network," ACM SIGCOMM, Philadelphia, Pennsylvania, Aug. 22-26, 2005.

[38]   R. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks," IEEE SNPA Workshop, Egan Convention Center, Anchorage, AK, May 11, 2003.

[39]   W. Zhao, M. Ammar, and E. Zegura, "A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks," MobiHoc, Ropping Hills, Tokyo, Japan, May 24-26, 2004.

[40]   Z. Zhang, "Routing in Intermittently Connected Mobile Ad Hoc Networks and Delay Tolerant Networks: Overview and Challenges," IEEE Communications Surveys Tutorials, Vol. 8, No. 1, pp. 24-37, 2006.

[41]   A. Jindal and K. Psounis, "Fundamental Mobility Properties for Realistic Performance Analysis of Intermittently Connected Mobile Networks," PerCom Workshop on Intermittently Connected Mobile Ad Hoc Networks (ICMAN), White Plains, NY, Mar. 2007.

[42]   Q. Zheng, X, Hong, P, Wang, L, Tang, and J. Liu, "Delay Management in Delay Tolerant Network," Int'l Journal of Network Management, 2008.

[43]   G. Karlsson, V. Lenders, and M. May, "Delay-Tolerant Broadcasting," ACM SIGCOMM Workshops, Pisa, Italy, Sep. 11-15, 2006.

[44]   S. Jain, K. Fall, and R. Patra, "Routing in a Delay Tolerant Network," ACM SIGCOMM Computer Communication Review, vol. 34, no. 4, pp. 145-158, Portland, OR, Aug. 30 – Sep. 3, 2004.

[45]   S. Merugu, M. Ammar, and E. Zegura, "Routing in Space and Time in Networks with Predicable Mobility," Georgia Institute of Technology, Technical Report, GIT-CC-04-07, 2004.

[46]   R. Handorean, C. Gill, and G. Roman, "Accommodating Transient Connectivity in Ad Hoc and Mobile Settings," Pervasive, Linz / Vienna, Austria, Apr. 18-23, 2004.

[47] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "RFC 4838, Delay-Tolerant Networking Architecture", IRTF DTN Research Group, Apr. 2007.

[48] A. Lindgren, A. Doria, and O. Scheln, "Probabilistic routing in intermittently connected networks," The Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2003.

[49] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for vehicle based disruption-tolerant network," IEEE INFOCOM, Barcelona, Catalunya, Spain, Apr. 23-29, 2006.

[50] J. Ghosh, H. Q. Ngo, and C. Qiao, "Mobility Profile based Routing Within Intermittently Connected Mobile Ad hoc Networks (ICMAN)," International Conference on Wireless Communications and Mobile Computing (IWCMC), Vancouver, Canada, July 3-6, 2006.

[51] Y. Wang and H. Wu, "DFT-MSN: The Delay Fault Tolerant Mobile Sensor Network for Pervasive Information Gathering," IEEE INFOCOM, Barcelona, Catalunya, Spain, Apr. 23-29, 2006.

[52] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "DTN routing as a resource allocation problem," ACM SIGCOMM, Kyoto, Japan, Aug. 27-31, 2007.

[53] K. Tan, Q. Zhang, and W. Zhu, "Shortest Path Routing in Partially Connected Ad Hoc Networks," IEEE Globecom, San Francisco, Dec. 1-5, 2003.

[54] E. P. C. Jones, L. Li, J. K. Schmidtke, and P. A. S. Ward, "Practical Routing in Delay Tolerant Networks, IEEE Transactions on Mobile Computing, Aug. 2007.

[55] Y. Gong, Y. Xiong, Q. Zhang, Z. Zhang, W. Wang, and Z. Xu, "Anycast Routing in Delay Tolerant Networks," Technical Report MSR-TR-2006-04, Microsoft Research, Jan. 2006.

[56] W. Zhao, M. Ammar, and E. Zegura, "Controlling the Mobility of Multiple Data Transport Ferries in a Delay-Tolerant Network," INFOCOM, Miami, FL, Mar. 13-17, 2005.

[57] Q. Li and D. Rus, "Communicating in Disconnected Ad Hoc Networks Using Message Relay," Journal of Parallel and Distributed Computing, 63, 2003.

[58] B. Burns, O. Brock, and B. Levine, "MV Routing and Capacity Building in Disruption Tolerant Networks," IEEE INFOCOM, Miami, FL, Mar. 13-17, 2005.

[59] P. Jackquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized Link State Routing Protocol for Ad Hoc Networks," IEEE International Multi –topic Conference, Lahore, Pakistan, Dec. 28-30, 2001.

[60] Y. Ge, Thomas Kunz, and Louise Lamont, "Quality of Service Routing in Ad Hoc Networks Using OLSR," 36th HICSS, Big Island, HI, Jan. 6-9, 2003.

[61] C. Perkins and E. Royer, "Ad hoc On-Demand Distance Vector Routing," 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, Feb. 1999.

[62] D. Johnson, D. Maltz, and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks," Ad Hoc Networking, 2001.

[63] Z. J. Haas, M. Perlman, "The Performance of Query Control Schemes of the Zone Routing Protocol" IEEE/ACM Transactions on Networking, vol. 9, no. 4, pp. 427-438, Aug. 2001.

[64] Y. Ko and N. Vaidy, "Location-Aided Routing in Mobile Ad Hoc Networks," ACM Wireless Networks Journal, June 2000.

[65] M. Grossglauser and M. Vetterli, "Locating Nodes with EASE: Mobility Diffusion of Last Encounters in Ad Hoc Networks," IEEE INFOCOM, San Francisco, CA, Mar. 30 - Apr. 3, 2003.

[66] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, "Age Matters: Efficient Route Discovery in Mobile Ad Hoc Networks using Encounter Age," Mobihoc, Annapolis, MD, June 1-3, 2003.

[67] J. J. Garcia-Luna-Aceves, M. Mosko, and C. Perkins, "A New Approach to On-Demand Loop-Free Routing in Ad Hoc Networks," 22nd ACM Symp. PODC, Boston, MA, July 13-16, 2003.

[68] M. Grossglauser and D. Tse, "Mobility Increases the Capacity of Ad Hoc Wireless Networks," INFOCOM, Anchorage, AK, Apr. 22-26, 2001.

[69] A. L. Iacono and C. Rose, "Infostation : New Perspectives on Wireless Data Networks," WINLAB, Technical Document, Rutgers Univ., 2000.

[70] R. Groenevelt, P. Nain, and G. Koole, "The Message Delay in Mobile Ad Hoc Networks." in *Proc. of PERFORMANCE,* October 2005.

[71] B. Liang and Z. J. Haas, "Predictive Distance-based Mobility Management for PCS Networks," *Proceedings of IEEE INFOCOM*, March 1999.