EFFICIENT BROADCAST IN MOBILE ADHOC NETWORKS UTILIZING RATELESS CODING

By

JASON DOUGLAS PITTS

Bachelor of Science in Electrical Engineering

Oklahoma State University

Stillwater, Oklahoma

2009

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2009

EFFICIENT BROADCAST IN MOBILE ADHOC NETWORKS UTILIZING RATELESS CODING

Thesis Approved:

Dr. Nazanin Rahnavard

Dr. George Scheets

Dr. Qi Cheng

Dr. A. Gordon Emslie

Dean of the Graduate College

ACKNOWLEDGMENTS

Jason Pitts thanks Dr. Nazanin Rahnavard for the support and mentorship she provided during the formation and research of this work. Further, he thanks Dr. Sohum Sohoni and the members of his computer architecture research groups, the CAESAR Lab, for their contributions of additional computing resources, without which this project would have taken many more months. Lastly, thanks go out to the entirety of the faculty at Oklahoma State University's School of Electrical and Computer Engineering, whose teaching and support all aided in the path to this work.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

CHAPTER I

INTRODUCTION

There is growing interest in the use of large-scale sensor networks for data collection/dissemination in a number of environments. Advances in technology have made the manufacture and deployment of these networks much more cost effective and thus more research is being conducted on more efficient communication methods for these networks. Many of these protocols rely on some means of one-to-all broadcast in order to disseminate information amongst the entire network, for instance as a means of establishing routes for point to point messages within the network or code update. Due to the omnipresence and reliance on broadcast protocols, much work has been done to develop efficient methods that maximize reachability (defined in chapter III) while minimizing energy requirements. In this thesis we propose and evaluate a broadcast protocol called mobile adaptive probabilistic broadcast (MAPcast), which is intended for efficient broadcasting in mobile adhoc networks (MANETs). In chapter II we will examine existing work in the field of wireless broadcast, as well as some of the methods against which MAPcast was compared to evaluate its merits. This chapter will also explain how rateless coding, a central element of the protocol, works. Chapter III describes in detail the development and final implementation of the MAPcast protocol. Chapter IV provides information of the performance of MAPcast as well as an evaluation of the protocol in comparison to existing methods of network wide broadcast. Finally, chapter V will present the concluding statements of this work and future work.

CHAPTER II

BACKGROUND

Chapter II introduces background material relevant to the discussion and evaluation of the MAPcast protocol. Sections II.A and II.B present information on existing forms of broadcast. Section II.C discusses rateless coding, a concept that is central to the performance of MAPcast. Section II.D discusses a few ideas that led to the development of MAPcast.

A. Flooding

Flooding [1] is widely used as a means of broadcast with high reachability (defined as the fraction of nodes in the network that are able to receive/decode the entirety of the original data). In this method each node rebroadcasts each packet it receives for the first time. This ensures that every node within the communication range of a transmitter receives a packet if the communication links are lossless. This method has been demonstrated to have the highest reachability in static networks; however, there is a high energy cost associated with it as every node retransmits each packet. Thus for a connected network of size (number of nodes) M, each packet will be transmitted M times. This level of retransmission creates large amounts of overlap in transmissions, which is inefficient. Energy inefficiency is not the only drawback to Flooding. The large number of transmissions can lead to the broadcast storm problem [2,3] as well, meaning that further retransmissions will be required as well as methods to deal with channel contention issues. Further the high reachability of the protocol in static networks does not necessarily carry over to mobile networks. This is due to the changing connections in the network over time. Fig. 1 shows an example situation. At time **A** the nodes establish a communication chain. In a static network with this layout the Node 1 could forward a packet to Node 2 which could then forward the packet to Node 3. However in a mobile network this chain is broken by at time **B**. In this scenario, Node 1 transmits to Node 2 at time **A**, but by the time Node 2 is ready to forward the packet, time **B**, Node 3 is no longer in range of Node 2. In fact it would be more beneficial for Node 1 to transmit twice, once at time **A** and again at time **B**.

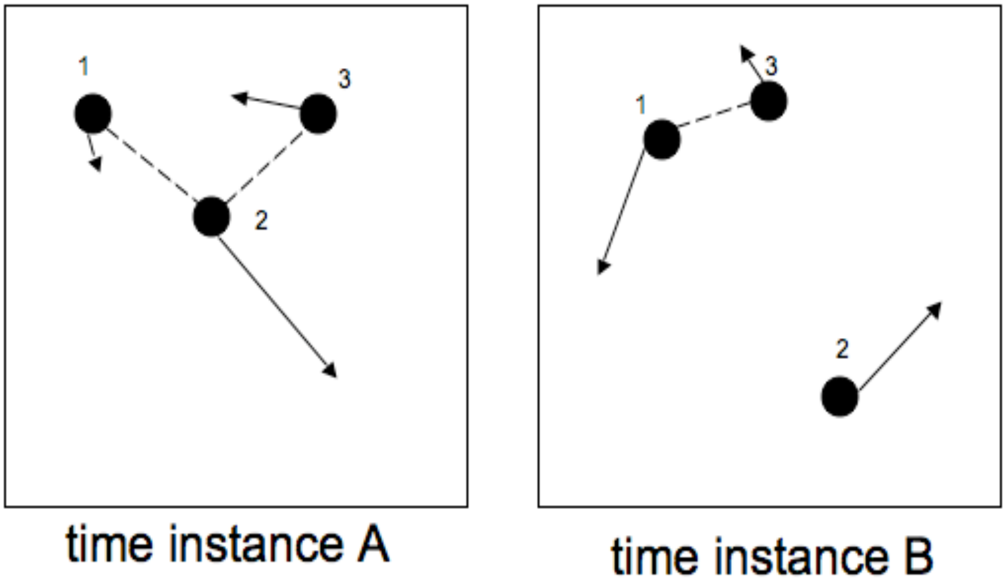

Figure 1. Node connections in mobile network - Dashed lines indicate communication links, arrows indicate a node's heading and relative velocity.

B. Probabilistic Broadcasts

MAPcast was developed using a probabilistic forwarding protocol. In probabilistic forwarding each node forwards received packets with a certain probability. In most models this probability is fixed and is common to all nodes. The value for this probability that optimizes the system (maximizes the reliability of the broadcast while minimizing the total number of transmissions) is highly dependent on the parameters of the network, such as node density relative to transmission range. This method is shown to provide good results [2]; however, it is not adaptable at run-time to changes in the network size. Another negative point to this method is that it makes an assumption of uniform node distribution. If a section of the network has a higher node density than the remainder of the network its optimal probability value is correspondingly lower than the rest of the nodes in the network. To alleviate this problem and to attempt to increase the efficiency of the broadcast dynamic assignment of forwarding probability has been proposed. Many methods of assignment of this probability have been proposed based on a number of parameters such as distance between receiver and transmitter, number of one-hop neighbors, and counter-based methods [2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17]. Dynamic forwarding probabilities based on distance attempt to assign a node's forwarding probability based on the distance between a receiving node and the node which transmitted the packet. This distance can be calculated using a locator device such as a GPS receiver or can be estimated using the received signal strength (RSS) value of the packet. The aim of this method is to give higher forwarding probabilities to nodes whose transmission will have a larger additional coverage area. As discussed earlier, the notion of additional coverage area is not as meaningful for mobile networks as it is for static networks, as the connectivity and location of nodes changes over time. Counter-based probability assignment has a large range of specific implementations, with the main goal of the majority being to assign a forwarding probability that is inversely related to the number of duplicate receptions of the packet to be forwarded. The idea being that if a node receives a high number of transmissions of a packet then the area in which the node residues has a high probability of being saturated with the packet. In other words there is a low number of additional nodes that will receive the packet which have not received it from the previous transmissions. To do this counter-based protocols wait a certain amount of time between reception of a packet and the forwarding of that packet. During this waiting period the node keeps track of how many duplicates of that packet are received beyond the first. This information is then used to assign the forwarding probability for the node.

C. Rateless Codes

Rateless codes (a.k.a. fountain codes) are a recent development in packet-level coding [18,19,20,21]. These codes allow a nearly unlimited number of unique encoded packets to be created for transmission. Rateless coding works by dividing the data into K packets each of length M. A random number D of these packets, where D has a distribution given by $\Omega(D) = \{\Omega_1, \Omega_2, \dots, \Omega_k\}$ where Ω_i is the probability that $D = i$, are combined to form an encoded packet. This D is known as the degree of the encoded packet. The packets chosen for this combining process are selected randomly from the original K packets. The random packets are bitwise XORed to produce the encoded packet. In this way a near unlimited number of unique encoded packets can be created provided that K is sufficiently large. This property is where rateless codes receive their name, because unlike traditional codes like the well known Hamming codes and LDPC codes, there is no set number of encoded symbols created from the data symbols to be transmitted. Rateless codes are sometimes known as fountain codes because they can send out an unlimited stream of symbols. In many implantations of these codes, the transmitter will continue to transmit encoded symbols until the receiving node notifies the transmitter to cease transmissions. To guarantee the successful decoding of a message encoded with a rateless code, a receiving node must receive an equal number or slightly more than K packets. In fact the receiver needs $\gamma \bullet K$ packets, where $\gamma \geq 1$. This γ is known as the overhead for the code. The relations between data packets and the encoded packets can be visualized as a bipartite graph with the nodes representing uncoded and encoded packets, and the edges between these nodes representing the combinations amongst them. Fig. 2 shows a small example of a rateless code, where $K = 10$ and $\gamma = 1$. Rectangular nodes correspond to original data symbols (packets or bits), and the circular nodes correspond to the encoded symbols. The edges linking the nodes indicate which of the data symbols each encoded symbol is composed of.

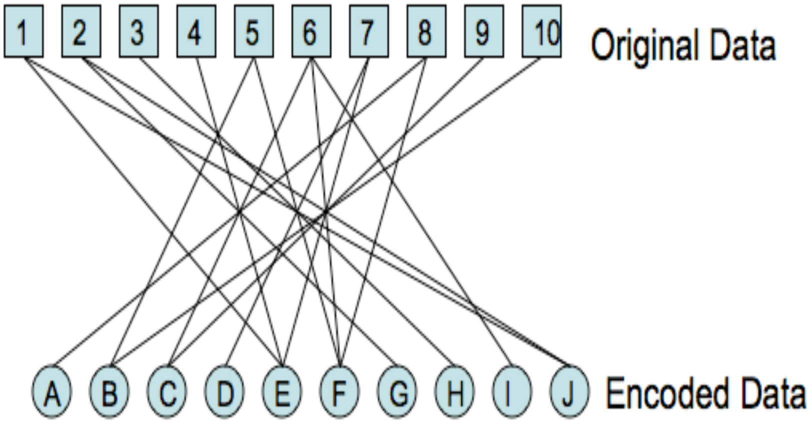

Figure 2. Bipartite graph representing the packet relationships in a rateless coding scheme.

Fig. 3,4,5 show decoding of a message with rateless coding. In the first stage of decoding, as shown in Fig. 3 each degree 1 encoded node is used to retrieve the information from its linked data node. In this example light blue data squares indicate unknown data values, dark blue squares indicate data values recovered on the current phase of decoding, and black square indicate data values recovered on a previous phase.

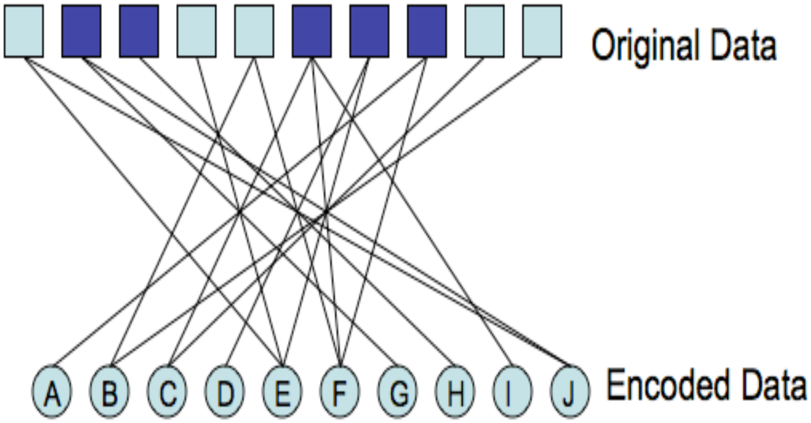

Figure 3. First decoding iteration.

Since these nodes are composed of only one data packet their value will be exactly the value of the associated data packet. For the second stage of decoding, shown in Fig. 4, the edges leading from known data packets are removed, while the connected encoded nodes' values are updated (XORed) with the data node. Then each degree 1 coded node is used to retrieve their data nodes.

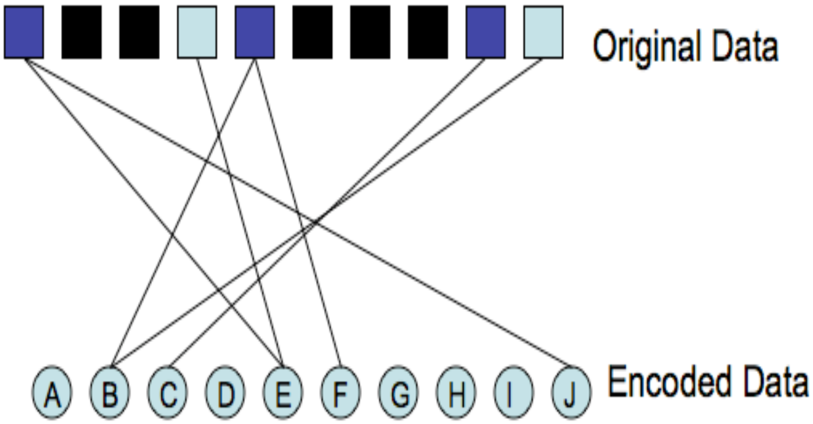

Figure 4. Second decoding iteration.

This process is repeated in Fig. 5 for a third round of decoding. In this example the decoding only required three iterations, however this will not necessarily be the case for much longer transmissions. The same process is used for decoding longer transmissions albeit with more decoding rounds. These codes work best with large K, which increases the uniqueness of each coded packet thus reducing γ. The gain in these codes comes from the fact that a node need not receive particular packets so long as it receives at least the γ·K encoded packets required for decoding. This works especially well with probabilistic forwarding methods, since this coding method can compensate for the lack of a particular packet's forwarding, a case which can be quite common in probabilistic forwarding methods.

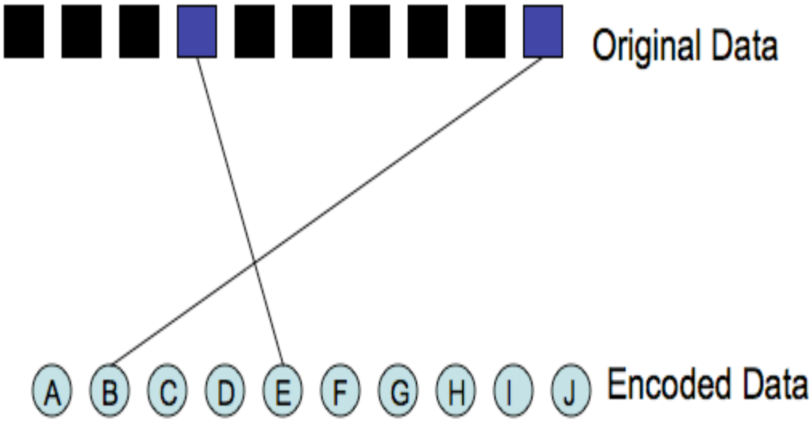

Figure 5. Final decoding iteration.

D. Evolution

The idea for MAPcast protocol was not created from scratch but rather was the end result of a progression of ideas. The beginning point of MAPcast was the desire to create a protocol that could take advantage of the mobility of a network to reduce the amount of energy required to deliver a broadcast message to the entire network. The application environment for this problem was envisioned as a large region with a few mobile nodes which wandered about the region collecting data. These nodes could be small robots that moved around a forest collecting data, or perhaps the nodes were small computers worn by soldiers or emergency personnel in a disaster zone. The primary assumption of the situation though is that the communication nodes do not have control over their movement. This assumption constrained the problem, by eliminating any methods that would seek to control the graph's connectivity by moving nodes into ideal locations for network saturation. These nodes are also assumed to be very low power, so transmission ranges would be small. This further reduces the connectivity of the graph because the nodes would have to be physically closer to each other in order to be connected. Finally the system was intended to be largely scalable. This led us to develop a protocol that would be decentralized, meaning that there could be no controller nodes that had information about the entire (or at least large portions of) the network. Also since the network is mobile the nodes' neighbors are constantly changing so methods that took into account information regarding multihop neighbors were deemed inefficient, since this data would quickly become outdated. It was concluded that the primary source of power inefficiency in a network such as this would be redundant transmission. Redundant transmissions in this sense means, transmissions which are received by nodes that have already received the same transmission from another node. This is because of the broadcast nature of wireless medium, where a packet sent by a node will potentially be received by all neighboring nodes. To alleviate this problem the idea of coverage area was investigated. In a static wireless network, a node which relays a message from another node, by necessity must transmit the message into an area that has already received it (assuming identical omni directional antennae). This is due to the overlap of their antenna patterns. In fact, the maximum additional area is given as:

$$A_{2MAX} = \left(\frac{\sqrt{3}}{2} + \frac{\pi}{3}\right) \bullet R^2 \cong 0.61 \bullet A_1, \qquad (1)$$

where R is the transmission range of the nodes. This situation can be seen in Fig 6.

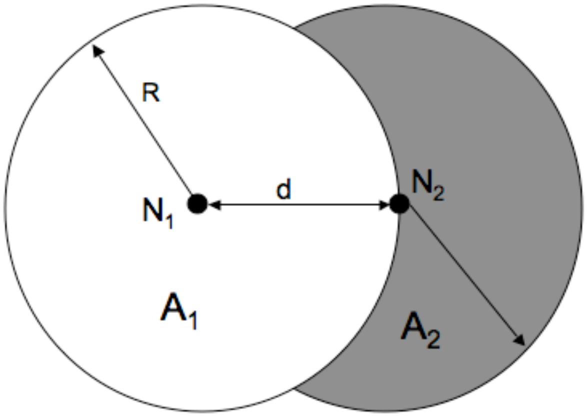

Figure 6. Maximum additional area (shaded region). A_2 is maximized when d is equal to R. A_2 is at most 61% of A_1.

The maximum additional area from a rebroadcast occurs when a node receives the message to be relayed at the extreme edge of the transmission radius. In this situation the relaying node only delivers the message to 61% of its transmission area that did not receive the message from the originating node. If the receiving node is not at the extreme edge the transmitters effective range then this region is even smaller, as seen in Fig 7.

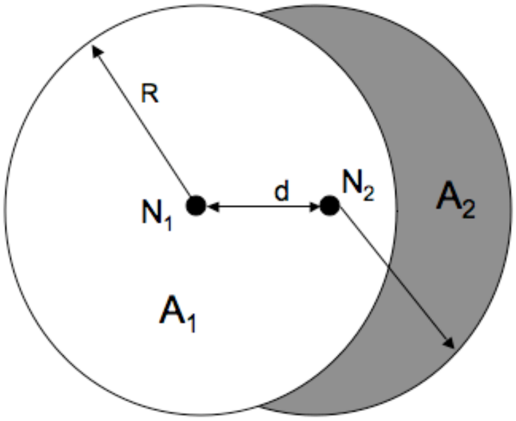

Figure 7. Additional coverage area. If d is less than R, then A_2 is less than its maximum value.

Adding to this the high probability that there are multiple relaying nodes in the originating nodes transmission range and the additional coverage area per relay transmission quickly decreases. This is one of the main causes for the development of probabilistic forwarding. This is the case for a static network, however for a mobile network this is not the case. In a mobile network this additional area is a non-deterministic function of time. If the relaying node is moving away from the point of origin of the message its additional coverage increases as time elapses, conversely if it is moving in the direction of the transmission it's additional coverage decreases as time elapses. However this additional area does not directly correlate to additional nodes which can be reached, as nodes can move into and out of areas which have already had a transmission. In order to maximize this additional coverage, we considered the idea of delaying transmission until a time in which the node has moved to a position wherein the entirety of its transmission area is outside of the transmission area of the originator of the message. To meet this objective we originally proposed two methods, both of which required some form of location awareness, such as a GPS receiver. The first method would have the relaying node transmit the message when it had traveled to a location that was 2*R away from the originating node (see Fig. 8).

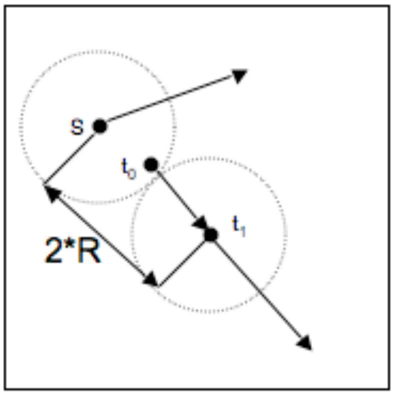

Figure 8. Transmit at 2R method. This method ensures that there is no overlap between the transmission areas of the two nodes.

This would require that as part of the packet structure, nodes include their location at the time of transmission. The second method still required that a node be aware of it's location but did not require the location of the originator. In this method the node would seek to transmit the message as far away as possible to ensure that another region of the network receives the packet. To this end the node would monitor its movement and transmit the message when it had moved as far along as its current path as it could. When the node detected a shift in direction of movement, the node would relay the message (see Fig. 9).

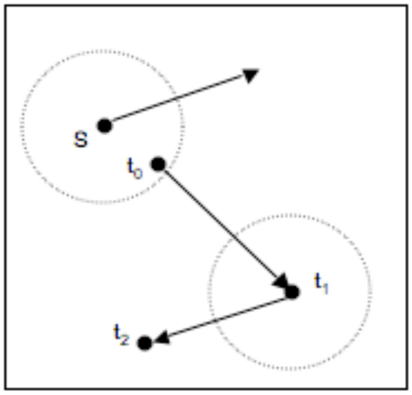

Figure 9. Transmit on direction change. This method forwards the packet when the node has traveled the entirety of its current heading.

These methods were explored but quickly discarded as they began to disregard one of the primary purposes of the protocol which was that all the nodes be low-cost and low-power, a requirement that was not inline with the inclusion of a GPS receiver. Also with the connectivity and location shifts of the graph over time there was not as much correlation between additional coverage area and additional nodes reached. In a static network these two are synonymous but this is not true for a mobile network. Thus the concept of additional nodes reached was directly explored.

Directly accessing the presence of neighbors as well as the packets they have collected requires some form of handshaking, usually in the form of beacon packets or packet queries. These interchanges between packets allow them to transmit which packets they have received as well as which packets they have not, or which they are requesting. The handshaking also helps to establish the presence of nearby nodes for purposes of determining the local node density. These handshaking packets are usually very small in comparison to the data packets that are being exchanged. This means that there is only a small amount of energy used in their transmission relative to the transmission of a data packet. Their small size also leads to the notion that the nodes are able to complete the handshake and still have time for data transmission in a mobile network. The nodes own mobility can create small windows in time in which the nodes are within each other’s communication ranges. If these handshaking packets are too long relative to the velocities of the nodes, the nodes can leave the effective communication range and be unable to profit from the information exchanged by the handshake. Having knowledge about the packets retained by a node’s neighbors allows it to tailor the transmission parameters (in our case whether or not to transmit) to be specific to the present case. This can create a great deal of adaptability since the transmission is not dictated by an overall or general case, but instead is set for the node’s specific “niche”.

The ideas presented in this section all contributed to the development of the MAPcast protocol. Their individual strengths and weaknesses were combined and modified to produce a broadcast method which sought to gain the best of each while making up for their shortcomings to produce a protocol with high reliability yet use less energy than other methods.

CHAPTER III

MAPCAST: AN EFFICIENT BROADCAST PROTOCOL FOR MOBILE ADHOC NETWORKS BASED ON RATELESS CODING

Chapter III presents the MAPcast protocol and explains the reason for certain design decisions.

A few notes on the vocabulary of this study:

Reachability/Reliability – These terms are used interchangeably in this thesis. It refers to the percentage of nodes in the network that receive all of the original data packets, either through receiving the actual packets (in methods not utilizing rateless coding), or through receiving at least $\gamma \cdot K$ encoded packets needed to decode (in methods with rateless coding).

Latency – In this work, latency is defined as the amount of time required to achieve 99.9% reachability. For methods that do not achieve full distribution of the broadcast, the latency is considered to be the time taken to reach its final reachability value.

Power consumption - In our model, we consider only the energy spent for RF transmissions as in [22]. Therefore, the energy consumption is proportional to the number of packet transmissions in the network.

MAPcast was developed as a power efficient method of broadcasting large data packets through a mobile ad-hoc network (MANET). Inspiration was taken from work done in the field of disruption tolerant networks (DTN), sometimes known as delay tolerant networks [23,24]. These types of networks are usually very sparsely populated with nodes that are not in constant contact with other nodes. MAPcast was designed to alleviate the problems associated with network scaling by being fully decentralized, meaning that there are no controlling nodes or storage of network information. Each node can only have (and only requires) up-to-date information about its one-hop neighbors. MAPcast uses a rateless coding scheme along with probabilistic retransmission to ensure high reachability/reliability along with better energy efficiency than other schemes. The combination works well due to the fact that rateless codes do not require specific packets, thus if a particular packet is not retransmitted into a sector of the network, due to the probabilistic nature of forwarding, it does not prevent nodes in that sector from successfully decoding the entire transmission. Nodes that receive the requisite number of packets for successful decoding can then decode the transmission and then re-encode to form new unique encoded packets for rebroadcast. In this manner the network can have multiple “sources” of unique packets. This increases the likelihood of packet transmission (as we will see), as it is unlikely that neighboring nodes will contain these newly generated encoded packets.

A. Scheduling

MAPcast is a schedule driven protocol, in which nodes only attempt transmission of packets at discrete times. This frees the node from constantly advertising the packets in its transmission buffer, thus reducing the amount of traffic in the network. The source begins transmitting its encoded packets in consecutive time intervals. It should be noted that while it is possible for the source to generate a near unlimited number of unique packets, it only transmits a set number $2K$, where K is the number of original data packets. This is due to the decentralized nature of MAPcast in which there is no reliable way for every node to inform the source when it has received enough packets to be complete. When a node receives a packet for the first time, it places an entry in its transmission schedule for a random time $t_{next\_attempt}$ time units in the future. This time is chosen as a Gaussian random variable with mean *WAIT_MEAN* and variance *WAIT_VAR*. These parameters can be adjusted based on the mobility of the nodes. This schedule entry also includes the number of times that the node will attempt to forward the packet. This value is initially set as *MAX_ATTEMPTS*, which also can be adjusted based on the mobility pattern. When the time for a transmission occurs, the node will send an ADV (advertisement) message to its neighbors. This ADV informs the neighboring nodes of what encoded packet is ready to be transmitted. The neighboring nodes will then reply with an ACK/NACK (acknowledge/negative acknowledge) message based on the following criteria.

1. If the neighboring node has not received the packet being advertised ***and*** the node has not received the number of packets required to successfully decode the transmission, the node will respond with an ACK message.
2. If the neighboring node has already received the packet being advertised ***or*** the node has received the number of packets needed to decode the transmission, the

node will respond with a NACK message.

The node wishing to transmit will then set the forwarding probability p based on the ratio of these received messages given by:

$$p = \frac{\#ACK}{\#ACK + \#NACK} \qquad (2)$$

The node will then retransmit the packet with probability p. If the packet is transmitted then it is removed from the transmitting node's schedule. In this case all the nodes which replied with an ACK place the packet in their schedule. If the packet is not transmitted (this happens with probability $1-p$), the node which attempted the transmission will then decrement the number of remaining transmission attempts by 1. If this number reaches 0 the schedule entry is then purged and the packet is said to have "died-out" for that node.

B. Recoding

If at any point in the simulation a node receives $\gamma \cdot K$ packets (the number required to decode) the node will immediately purge its schedule and begin attempting to decode the message and generating new encoded packets by recoding, an operation which requires approximately T_d time units. After this time the node will begin transmitting new encoded packets in consecutive time intervals. These secondary source nodes will only create K new packets, rather than the 2*K packets that the primary source created. These packets will also have *MAX_ATTEMPTS* attempts remaining; however due to their nature as freshly encoded packets they have a probability near 1 of being transmitted on their first attempt, so long as there is a node in the vicinity to receive the packet. This addendum to the protocol increases the amount of unique and diverse packets in the network thus increasing the probability of successful transmission and reliability of the network while decreasing the latency of the broadcasting.

C. Setting Parameters

The parameters for MAPcast were decided based on experimental results. The following figures show the effects of varying the *MAX_ATTEMPTS* and *WAIT_MEAN*. These data sets were averaged for five data sets for each variation of the parameters. Fig. 10a and 10b show the reliability and power usage, respectively, for MAPcast for different numbers of *MAX_ATTEMPTS*. Similarly Fig. 10c shows the latency of MAPcast for varying numbers of *MAX_ATTEMPTS*.

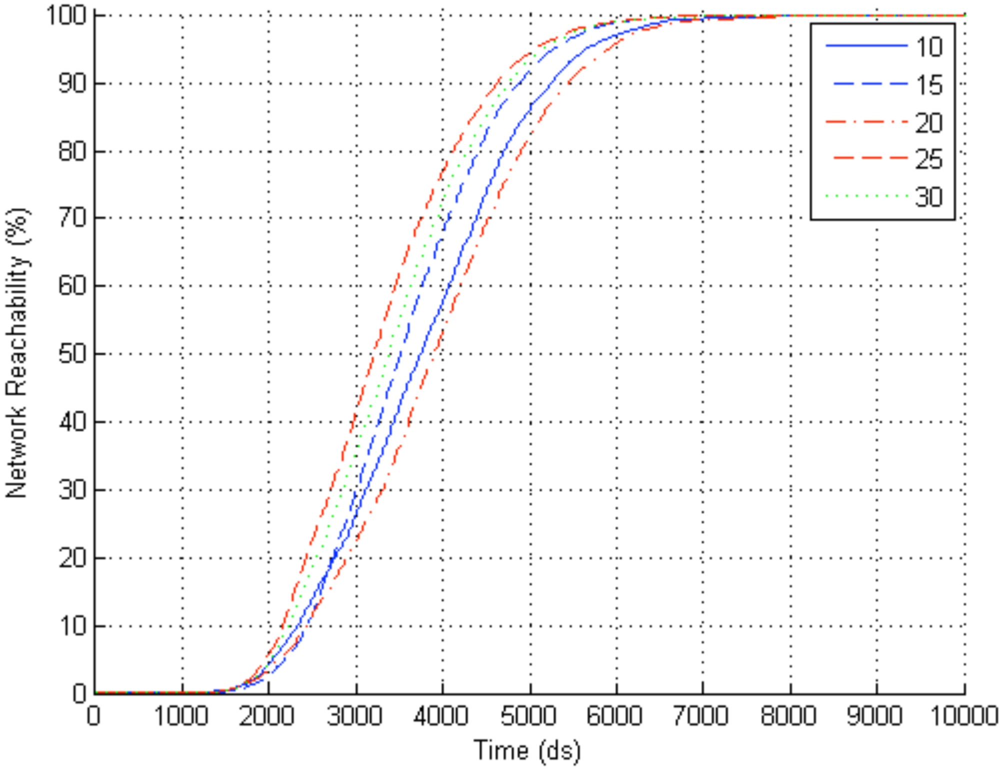

Figure 10a. Reachability - MAPcast: WAIT_MEAN set at 200. MAX_ATTEMPTS allowed to vary.

It can be seen in these figures that while all variations allow for full reachability, the version using 20 for *MAX_ATTEMPTS* consumes slightly less power. Additionally Fig. 10c indicates that there is not a strong correlation between number of attempts and latency.

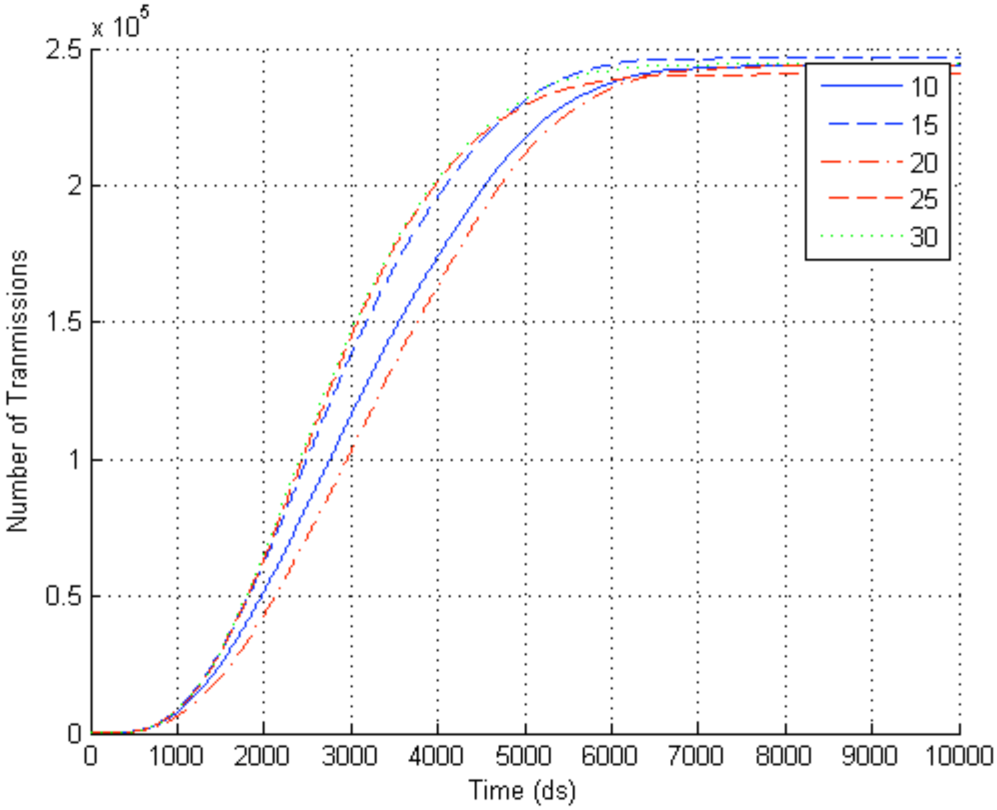

Figure 10b. Power Consumption - MAPcast: WAIT_MEAN set at 200 ds. MAX_ATTEMPTS allowed to vary.

Fig. 11a, 11b, and 11c show the performance of MAPcast while varying the WAIT_MEAN parameter. These data sets were taken with a fixed WAIT_VAR, the variance of the waiting period, of 30 deciseconds and a fixed MAX_ATTEMPTS of 20. Fig. 11a indicates that all variations allow for complete reachability. However as seen in Fig 11b, the variations that have longer wait times (i.e. 200 and 300) have lower power consumption.

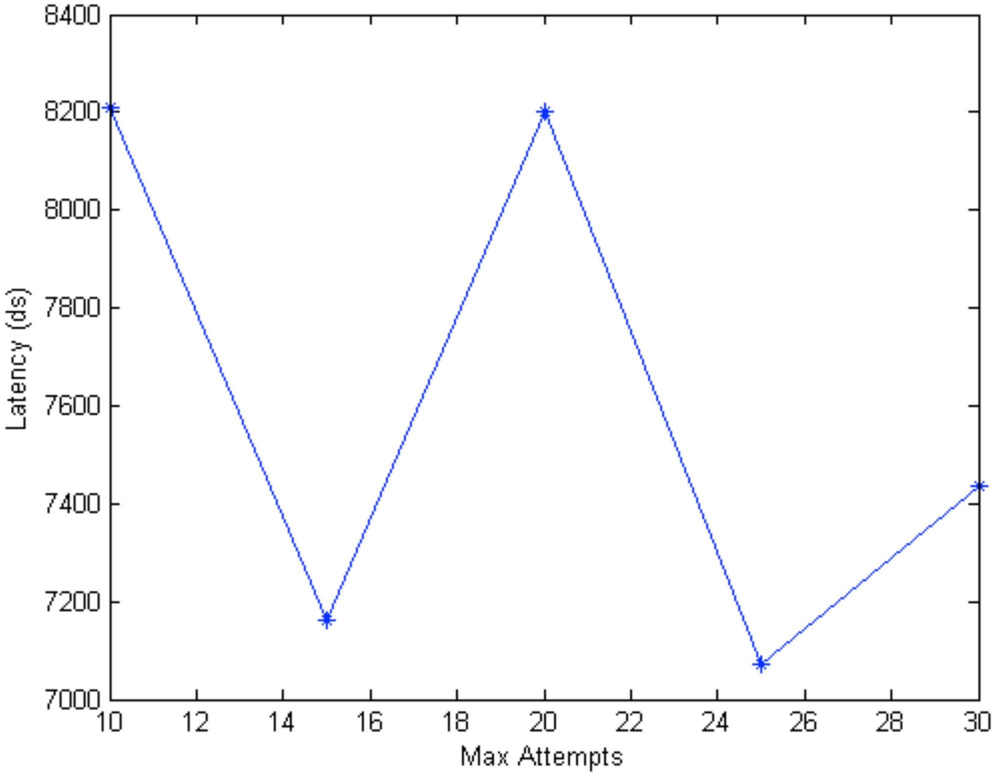

Figure 10c. Latency - MAPcast: WAIT_MEAN set at 200 ds. MAX_ATTEMPTS allowed to vary.

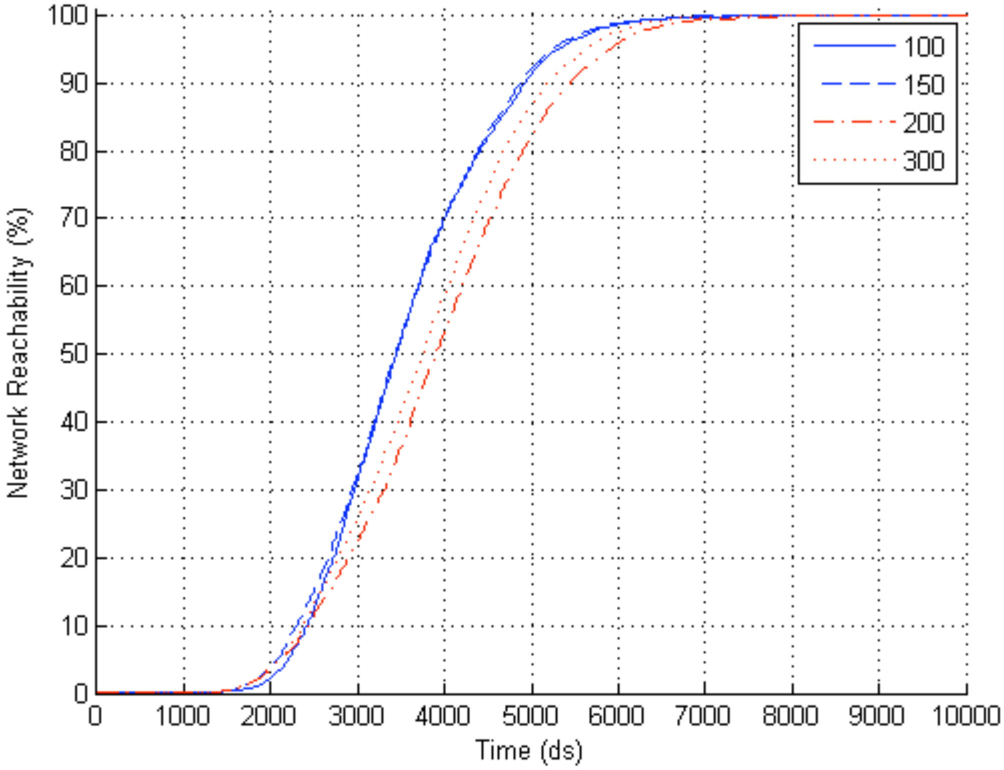

Figure 11a. Reachability - MAPcast with *MAX_ATTEMPTS* set at 20. *WAIT_MEAN* allowed to vary.

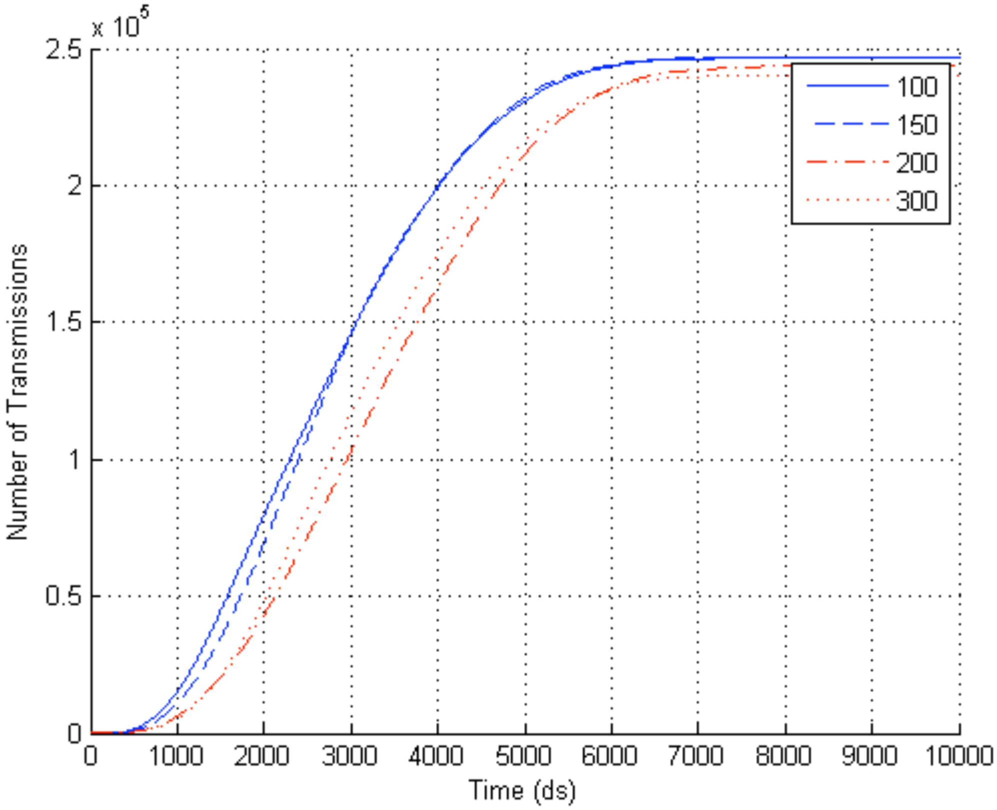

Figure 11b. Power Consumption - MAPcast with MAX_ATTEMPTS set at 20. WAIT_MEAN allowed to vary.

Additionally, there is little correlation between the mean wait time and the latency for these ranges of the parameters as seen in Fig. 11c.

The final parameters were decided upon as the best trade off between ensuring reachability and minimizing power consumption. Latency was also considered, but was not as heavily weighed in the decision making process.

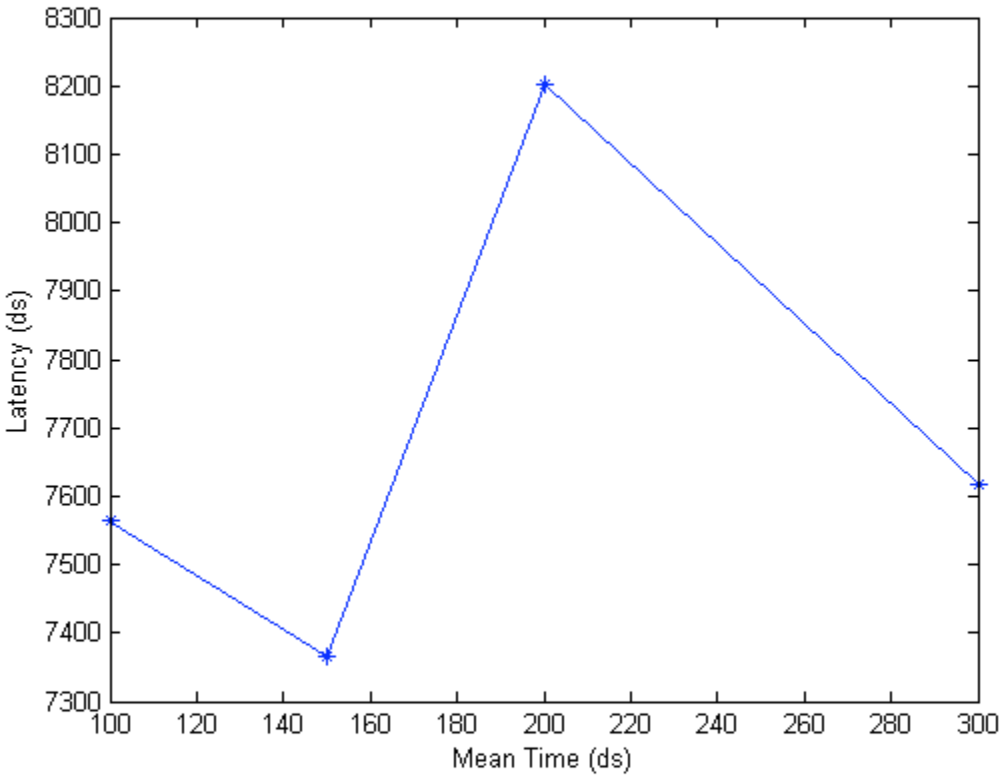

Figure 11c. Latency - MAPcast with MAX_ATTEMPTS set at 20. WAIT_MEAN allowed to vary.

D. MAPcast Protocol

A MAPcast broadcast begins with the source node encoding a number of packets. This set produces 2*K encoded packets from the original K data packets. The source, which is mobile node, then loads these packets into its transmission schedule in consecutive time slots with MAX_ATTEMPTS transmission attempts left. This number of attempts may seem unnecessary, as there is no way for the source's neighbors to send NACK packets, since no other node could have transmitted a packet that the source has not already. There is, however, the possibility that the node has **no** neighbors for a particular transmission attempt. If this were the case then the packet would "die out" and never be injected into the network if the source had only one attempt at its transmission.

The source begins making its transmission attempts. The likewise any forwarding nodes attempt to transmit their scheduled packets at the scheduled time step. In the event that two packets are scheduled at the same time, the packet placed in the schedule first is attempted and the other(s) is attempts on at the next time. When a transmission attempt is made the transmitter broadcasts an advertisement packet (ADV) to nodes within range. These nodes reply with the appropriate ACK/NACK message as described above. The node attempting to transmit then sets the probability of forwarding as shown in Eq. (2). If the transmission does in fact occur, the neighboring nodes which replied with an ACK place the packet in their transmission schedule at the Gaussian random time from the present with MAX_ATTEMPTS attempts remaining. If the transmission did not occur the attempting node reduces its remaining attempts and reschedules for a time in the future which is chosen from the Gaussian random set. Having these times be random rather then set times allows the network to have a chance to exploit many different time steps. Also by not transmitting at set times, the strain on the network from a large number of nodes attempting to transmit at the same time is reduced. This situation would occur when many nodes that received a packet have all scheduled its forwarding for the same time instant in the future, a situation that occurs if some of those nodes moved in the same direction at similar speeds. While MAPcast does not directly address MAC layer contention issues, they were considered while developing the protocol. If a packet's remaining attempts is reduced to zero the packet is removed from the schedule.

This process continues with nodes decoding and recoding where appropriate as explained in Section III.B. The protocol does not require that nodes somehow convey the fact that they have received the entire original data, like some methods, especially those utilizing rateless coding. This is due to the fact that as more of the nodes of the network receive the requisite $\gamma \cdot K$ packets needed to decode, the probability of forwarding packets decreases and the increasing numbers of packets "die-out" from the network.

Fig. 12 depicts the flowchart of the MAPcast protocol. Fig. 13a, 13b, and 13c depict the propagation of the broadcast through the network. Each snapshot of the propagation is separated by 500 deciseconds. We can see that by time 8000 ds, the entire network has received the broadcast. See Table I for the parameters of the network.

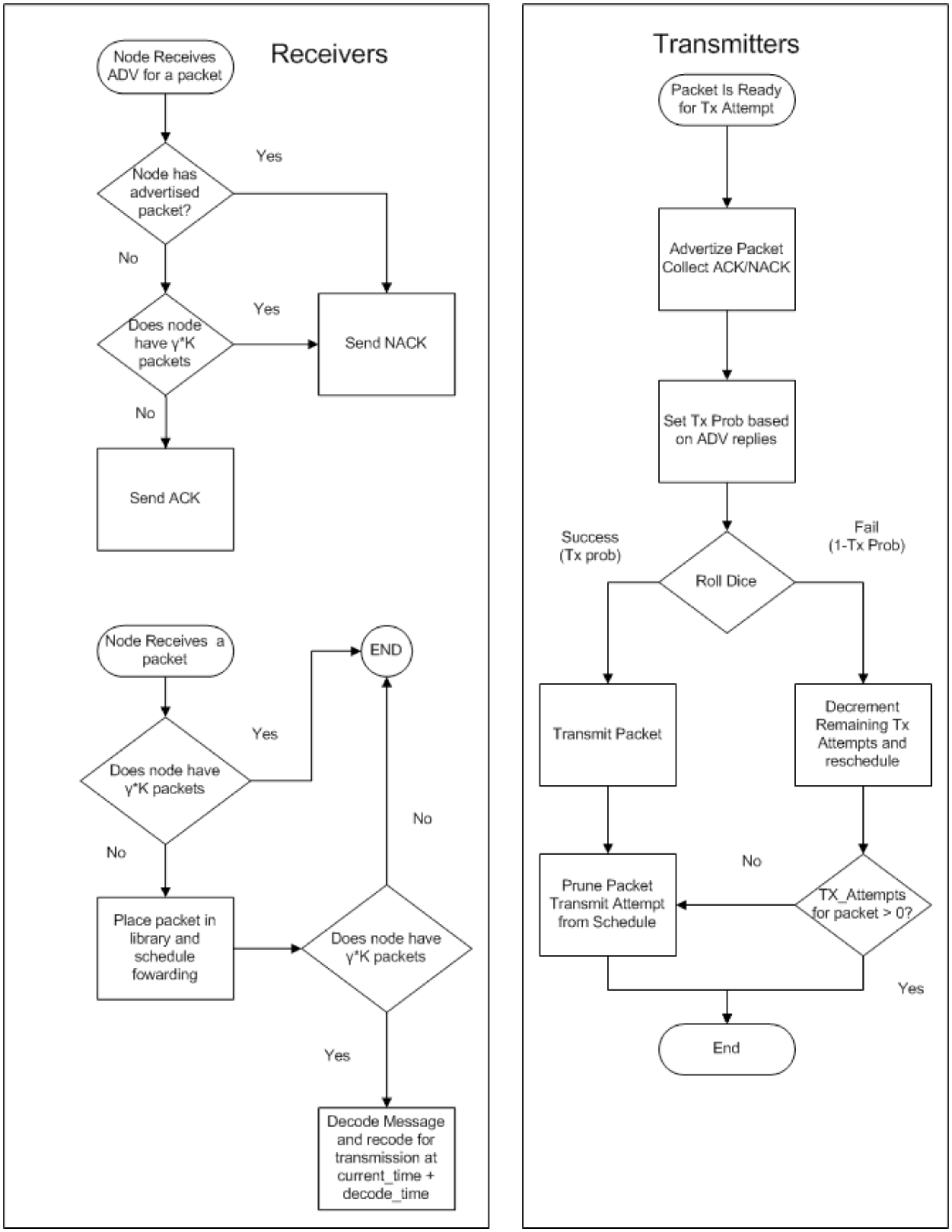

Figure 12. Flowchart of the MAPcast Protocol

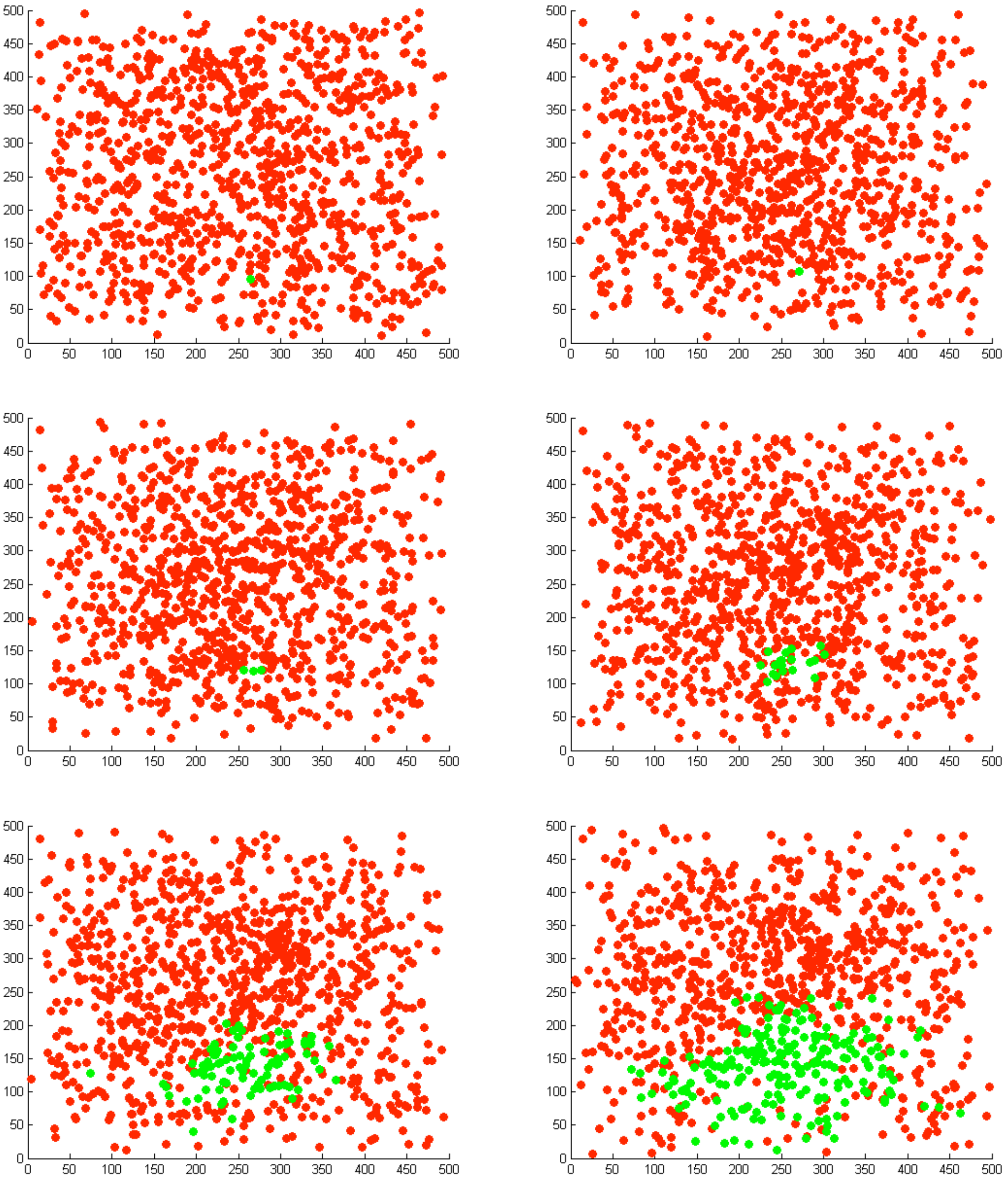

Figure 13a. Data propagation of MAPcast in a 1000 node network. Red nodes indicate nodes that have not received γK packets. Green nodes are nodes that have received γK packets. Snapshots are in 500 ds intervals. (500ds-3000ds)

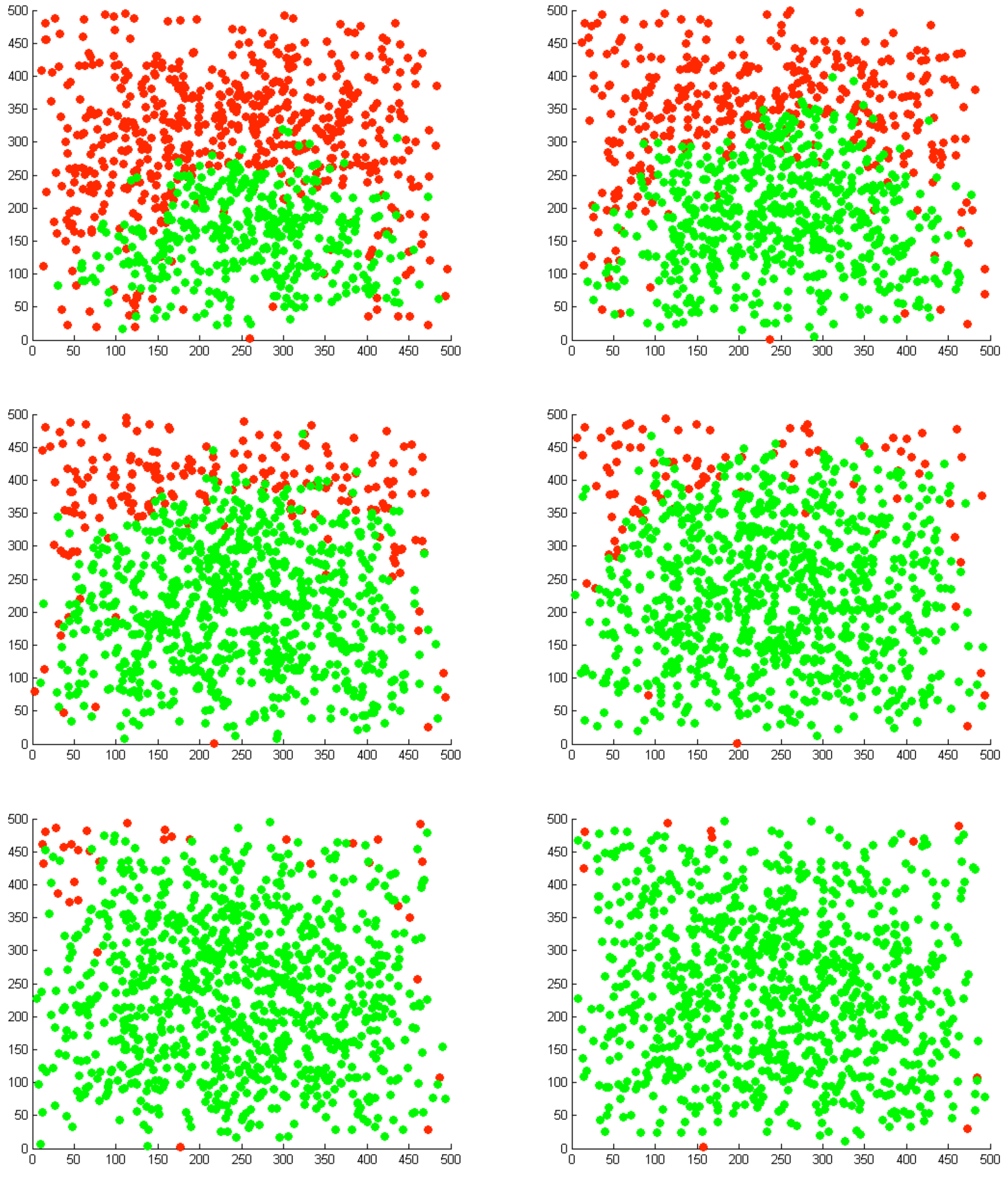

Figure 13b. Data propagation of MAPcast in a 1000 node network. Red nodes indicate nodes that have not received γK packets. Green nodes are nodes that have received γK packets. Snapshots are in 500 ds intervals. (3500ds-6000ds)

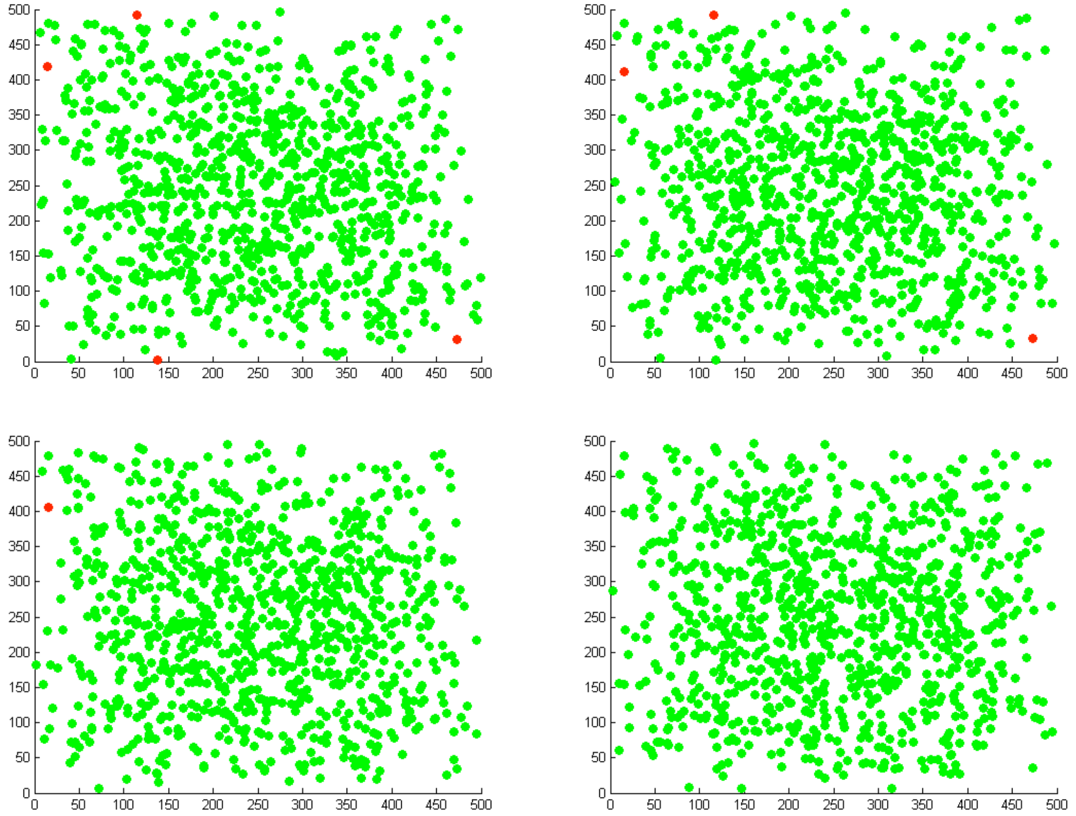

Figure 13c. Data propagation of MAPcast in a 1000 node network. Red nodes indicate nodes that have not received γK packets. Green nodes are nodes that have received γK packets. Snapshots are in 500 ds intervals. (6500ds-8000ds)

CHAPTER IV

SIMULATION RESULTS

Chapter IV presents the results obtained from simulating the MAPcast protocol in a mobile network. These results are compared to existing methods of broadcast, especially focusing on other methods of probabilistic broadcast. Additionally, these methods were modified to include rateless coding. In this way the performance of MAPcast can be examined while isolated from the gains of rateless coding on its own.

A. Mobility Pattern

For the simulations the node mobility was created using the random waypoint model. In this model nodes are initially distributed in a uniformly random manner within the bound of the field *F,* which has dimensions *Xmax* by *Ymax*. Each node uniformly at random selects a destination location within the field of the simulation ($X \in [0,Xmax]$, $Y \in [0,Ymax]$) , and then moves to the location with velocity *v*, which is also uniformly distributed ($v \in [0,Vmax]$). Each node then moves directly toward its destination at its selected velocity. When a node reaches its destination it selects a new destination and velocity in the same manner until the end of the mobility trace at *T* time units. Table I depicts the parameters and their corresponding values chosen for our simulations.

TABLE I

MOBILITY TRACE PARAMETERS

Variable	Symbol	Value
Field width	*Xmax*	500 (m)
Field height	*Ymax*	500 (m)
Max. Velocity	*Vmax*	2 (m/s)
Max. Time Units	*T*	10000 ds (deci second)
Number of Nodes	*M*	1000

B. MAPcast Parameters

Due to the stochastic nature of the protocol, the system is simulated a number of times which are then averaged out to give a good statistical look at the data. This is to eliminate variation due to the source of the broadcast (which is randomly chosen for each simulation run) as well as the probability based nature of the packet forwarding. For MAPcast, the simulation was run ten times for each variation of the simulation parameters. For the final version of MAPcast the following parameters were chosen as shown in Table II.

- *K* – the number of uncoded packets to be transmitted
- *N* – the number of coded packets the source node transmits
- γ - rateless code overhead – percentage of *K* packets required for decoding
- T_d – number of time units required to decode message and encode new packets
- *MAX_ATTEMPTS* – the number of times a node will attempt to transmit a given

packet

- *WAIT_MEAN* – mean time units before next transmission attempt for a given packet in the schedule
- *WAIT_VAR* – variance of time units before next transmission attempt
- *R* – maximum transmission range for each node

TABLE II

MAPCAST PARAMETERS

K	1000
N	2000
γ	1.03
Td	200 (ds)
MAX_ATTEMPTS	20
WAIT_MEAN	200 (ds)
WAIT_VAR	50 (ds)
R	25 (m)

C. Comparison Protocols

For the comparison protocols, the following parameters were used that are common to all methods. Methods utilizing rateless coding as well had the same K, N and γ as MAPcast. Also all comparison protocols, except distance-based dynamically adjusted probabilistic forwarding (DDAPF) [14], have one attempt at transmission and only one time unit wait between packet reception and transmission attempt. Methods that utilize packet decoding and retransmission use the same T_d as MAPcast.

DDAPF [14] is an adaptive probabilistic forwarding method proposed by Khan et al.

that uses exponential functions to set forwarding parameters. The method sets a forwarding probability based on the distance *d* between sender and receiver as given by:

$$p = \left(\frac{e - e^{\left(1 - \frac{d}{R}\right)}}{e - 1} \right)^k \qquad (3)$$

where R is the transmission range.

There is also a waiting period for retransmission that is similarly based on this distance. Our implementation of the protocol uses a *k* ([14]'s notation) value of 1, and a *Max_delay* of 10 (ds). Another version of this protocol was tested that uses rateless coding (called rateless-DDAPF) as well.

D. Simulation Results

MAPcast was tested against Flooding and fixed probabilistic methods of broadcasting that did not include rateless coding. These methods were found to have extremely low reachability, compared to MAPcast, and so were not included in the results section of this thesis. Therefore, in this thesis we integrated Flooding and PBcast with rateless coding to improve their performance.

Fig. 14a shows the network reachability of MAPcast as well as the reachability of a flooding transmission using rateless coding.

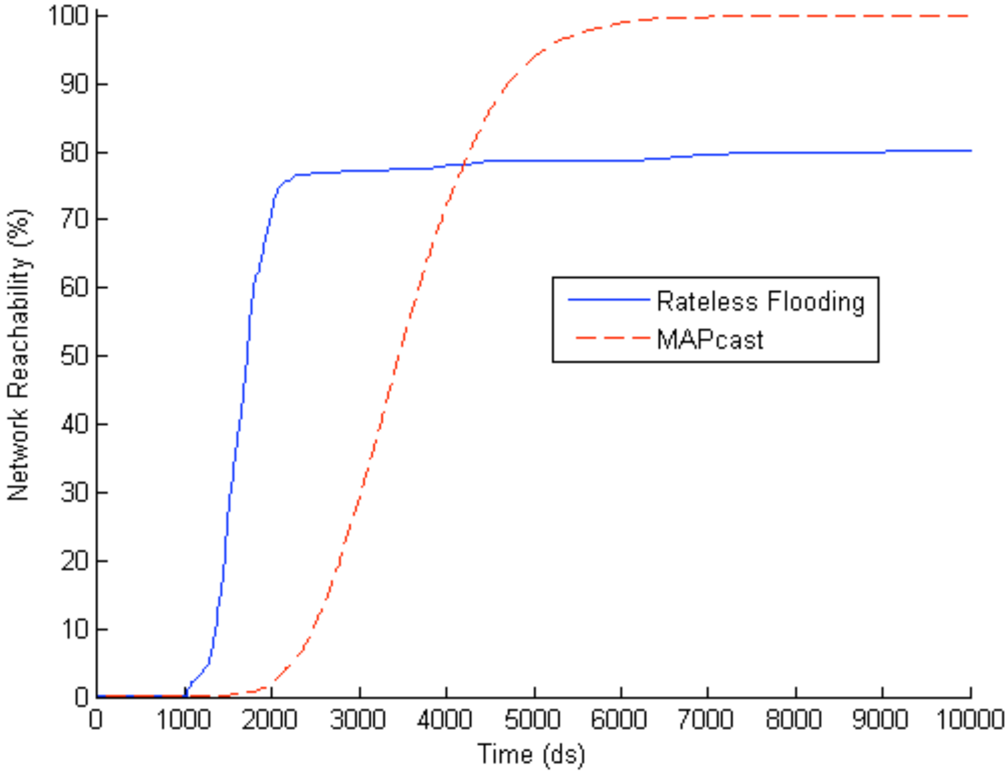

Figure 14a. Reachability - MAPcast vs. Rateless-Flooding

Flooding with rateless coding (call it Rateless-Flooding) is a method in which the source node encodes the data and transmits a number of these encoded packets. The other nodes in the network forward these packets with $p = 1$. It is evident from the Fig. 14a that MAPcast has a large gain over Rateless-Flooding in terms of reliability, in that it is able to deliver the message to the entire network, while Rateless-Flooding cannot. This is due to the fact that MAPcast has multiple attempts to transmit as well as a wait time between transmission attempts. This enables the nodes to travel a bit, and thus the connectivity graph of the network changes between the time of reception and the time of transmission. Therefore if there are any regions of the network that are partitioned (disconnected) from other regions of the network, the nodes have a chance to establish connectivity with the portioned regions. Also the multiple transmission attempts adds the ability to only forward in “good” regions (areas that have a high portion of nodes without the packet), this decreases superfluous transmissions.

In Fig. 14b we see the difference in power consumption between MAPcast and Rateless-Flooding. It shown in the figure, in terms of power consumption, MAPcast uses only about 20% of Rateless-Flooding’s power, which is a significant reduction of power consumption.

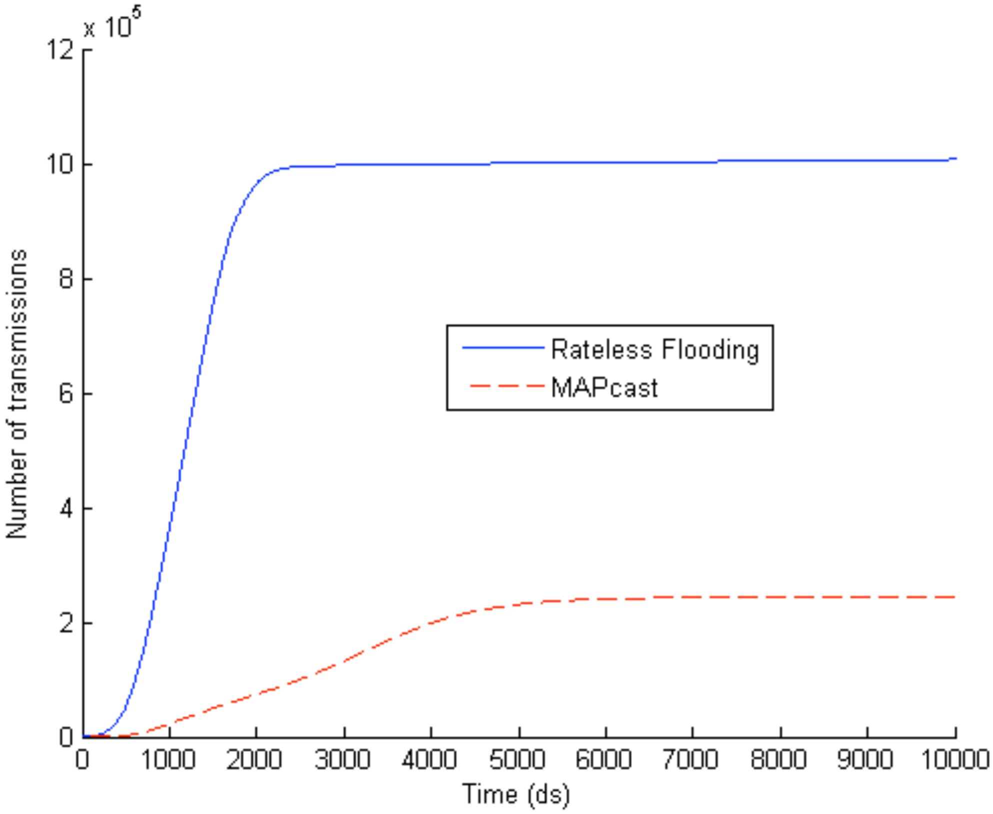

Figure 14b. Power Consumption - MAPcast vs. Rateless-Flooding

Next we see MAPcast compared with a probabilistic broadcast (PBcast) model with forwarding probability *p* that uses rateless coding at the source only. Fig. 15a and 15b show the reachability and power usages, respectively for different values of *p*. Again we see that even at the highest probability, Rateless-PBcast is unable to achieve the reliabilities that MAPcast can. Similarly, we see that for all but the lowest forwarding probabilities (*p*=0.3), MAPcast uses lower energy to achieve its reachability. It should be noted though that rateless-PBcast with *p*=0.3 has a very low reachability.

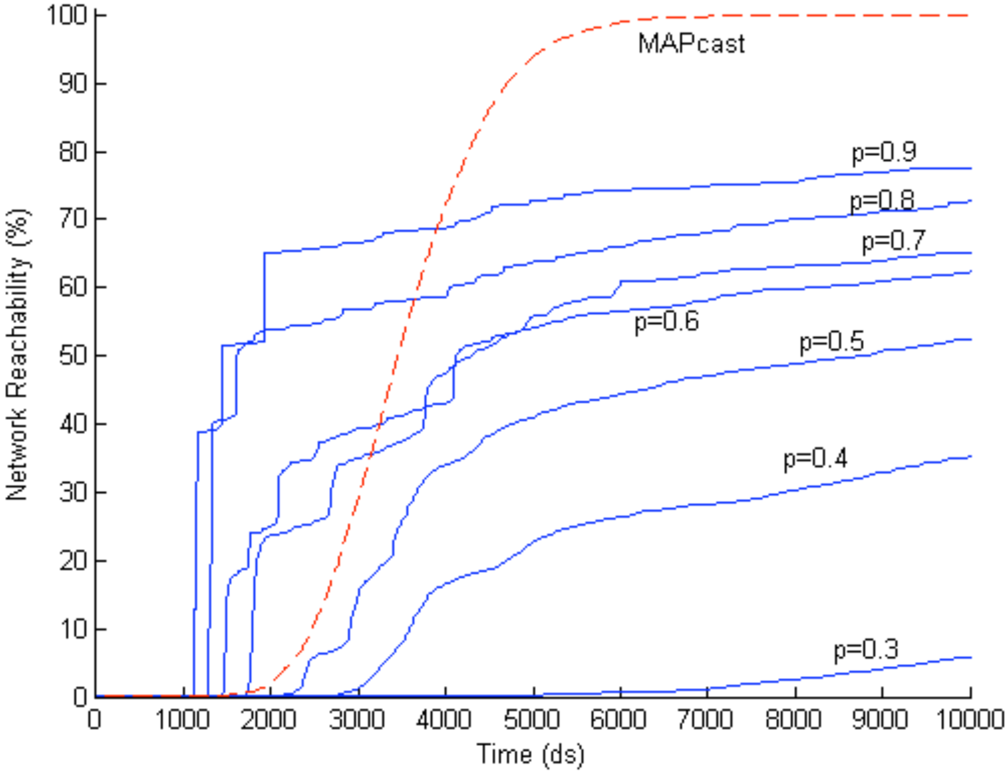

Figure 15a. Reachability - MAPcast vs. Rateless PBcast with forwarding probability *p*.

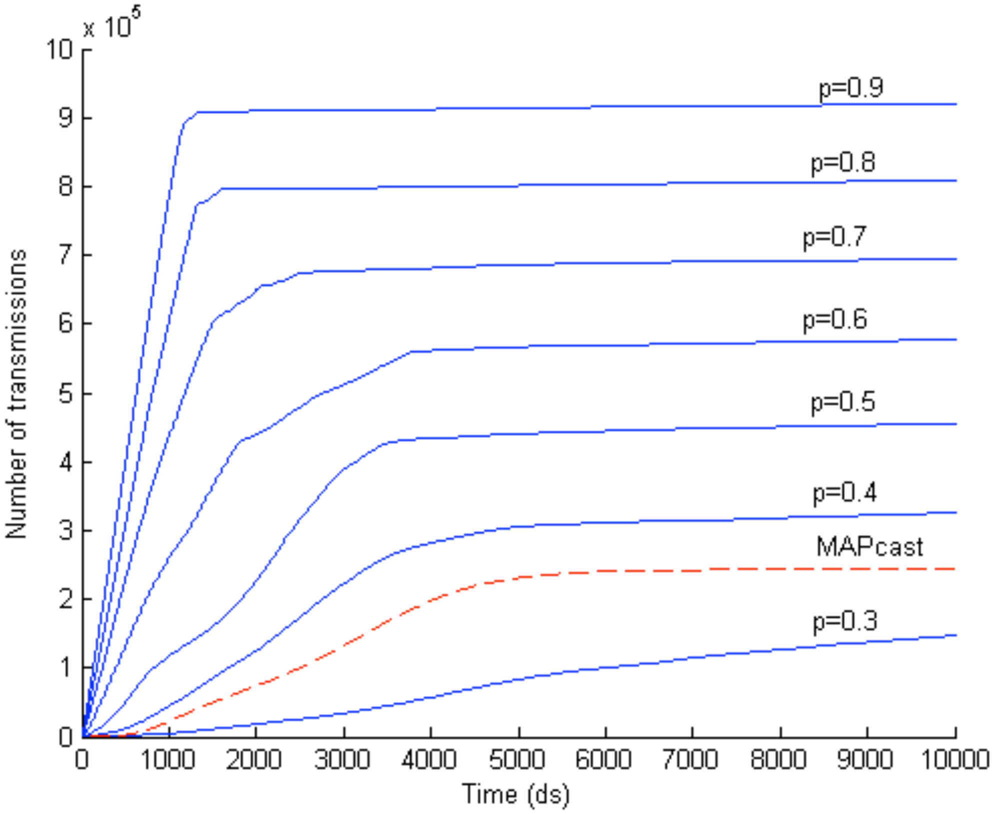

Figure 15b. - Power Consumption - MAPcast vs. Rateless PBcast with forwarding probability *p*.

In Fig. 16a and 16b the performance of MAPcast is evaluated against the DDAPF protocol described in [14]. Fig. 16a shows the reachability of the two protocols. Notice that DDAPF has a much quicker time to asymptote than MAPcast. This is due to the much shorter retransmit delay in DDAPF; however, final reachability of MAPcast is higher than that of DDAPF. Fig. 16b shows the power consumption of the protocols. Again we see a huge power savings in MAPcast.

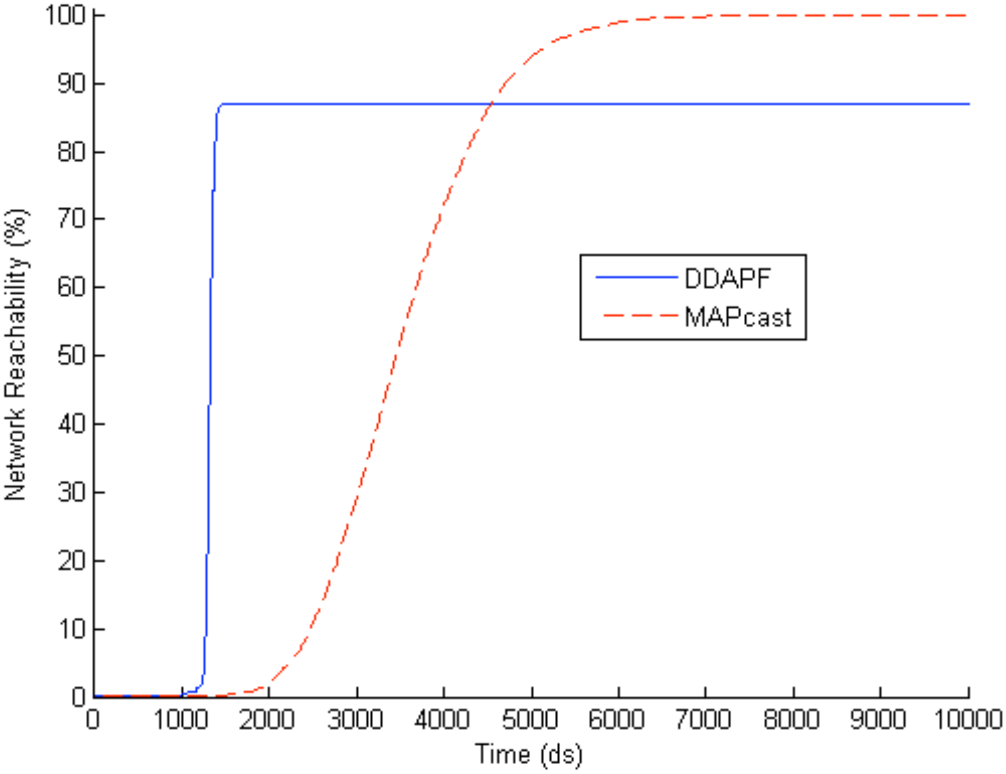

Figure 16a. Reachability - MAPcast vs. DDAPF

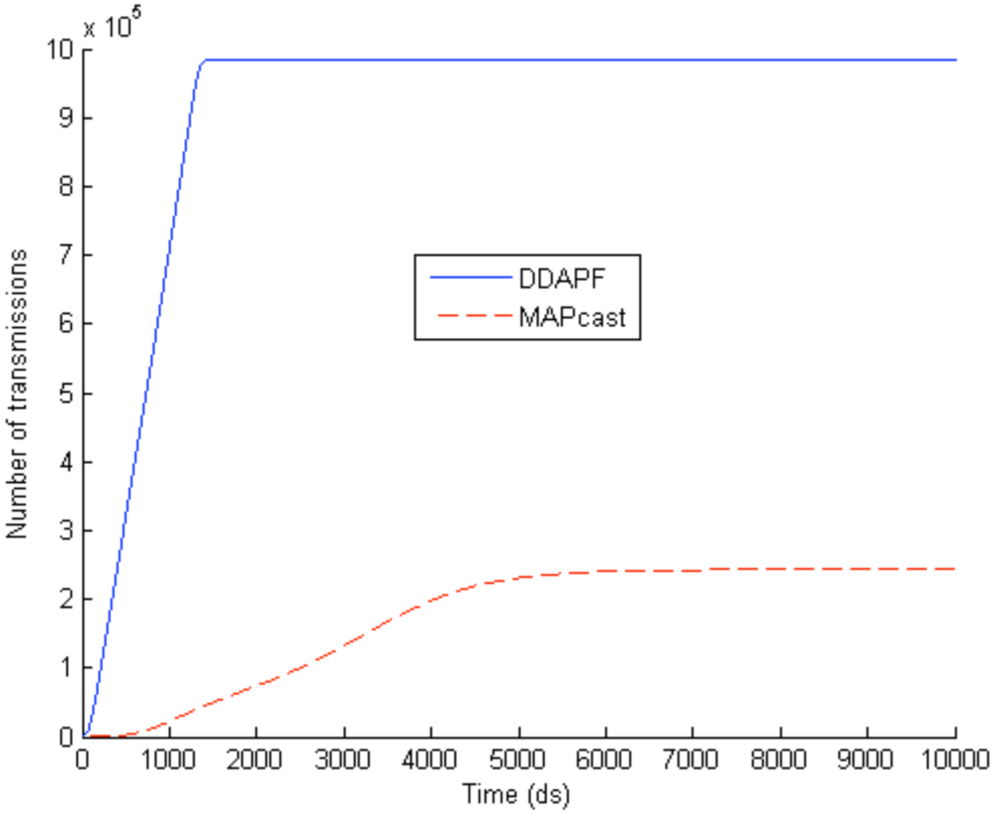

Figure 16b. Power Consumption - MAPcast vs. DDAPF

MAPcast was also compared to a version of DDAPF that was modified to use rateless coding. This was done to ensure that the gains achieved by MAPcast over DDAPF are not simply due to rateless coding, but are indicative of the superiority of MAPcast's forwarding characteristics. Fig. 17a shows the reliability results for Rateless DDAPF along with regular DDAPF (as in [14]) and MAPcast for comparison. From this figure we can see that the inclusion of rateless coding allows the network to fully distribute the data to the entire network (reachability of 100%). Also the latency is very low. However, in Fig. 17b it can be seen that the power consumption is very high, on the order of 17 times greater than MAPcast if we consider the power usage at the point that final reliability is reached. (Approximately 2900 ds).

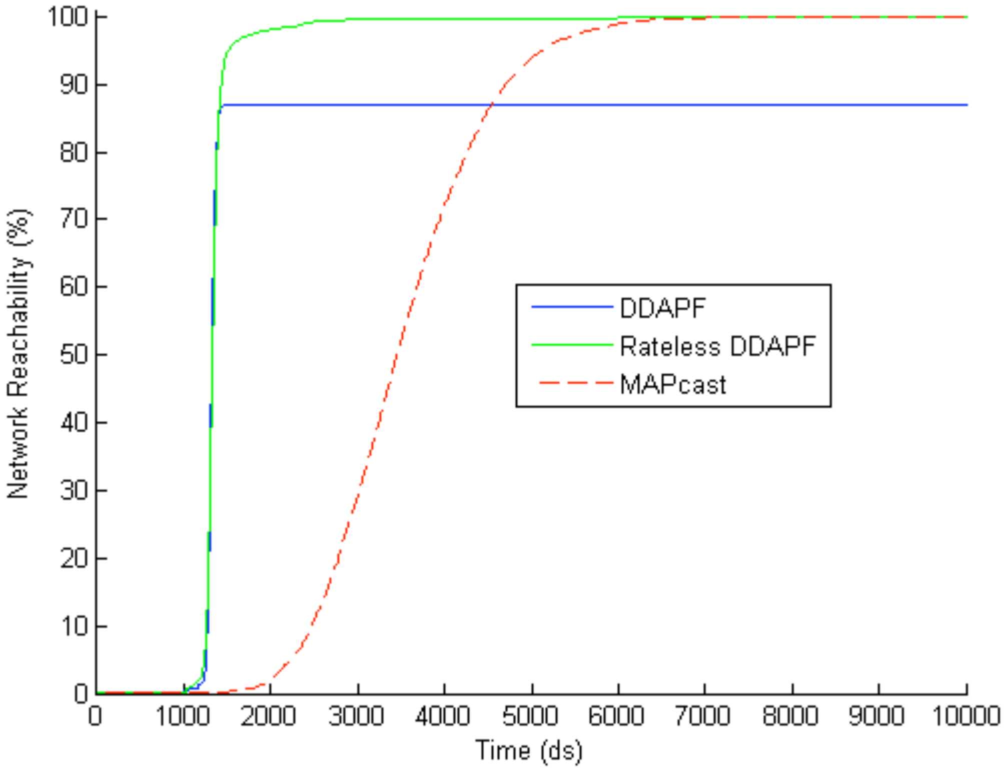

Figure 17a. Reachability - MAPcast vs. DDAPF with Rateless coding

In Fig 17b it can be seen that Rateless-DDAPF continues to transmit packets after all nodes in the network have received γ·K packets. This is because the Rateless-DDAPF has no method to notify the nodes that transmissions are no longer necessary. This problem is taken care of in MAPcast by the ADV system.

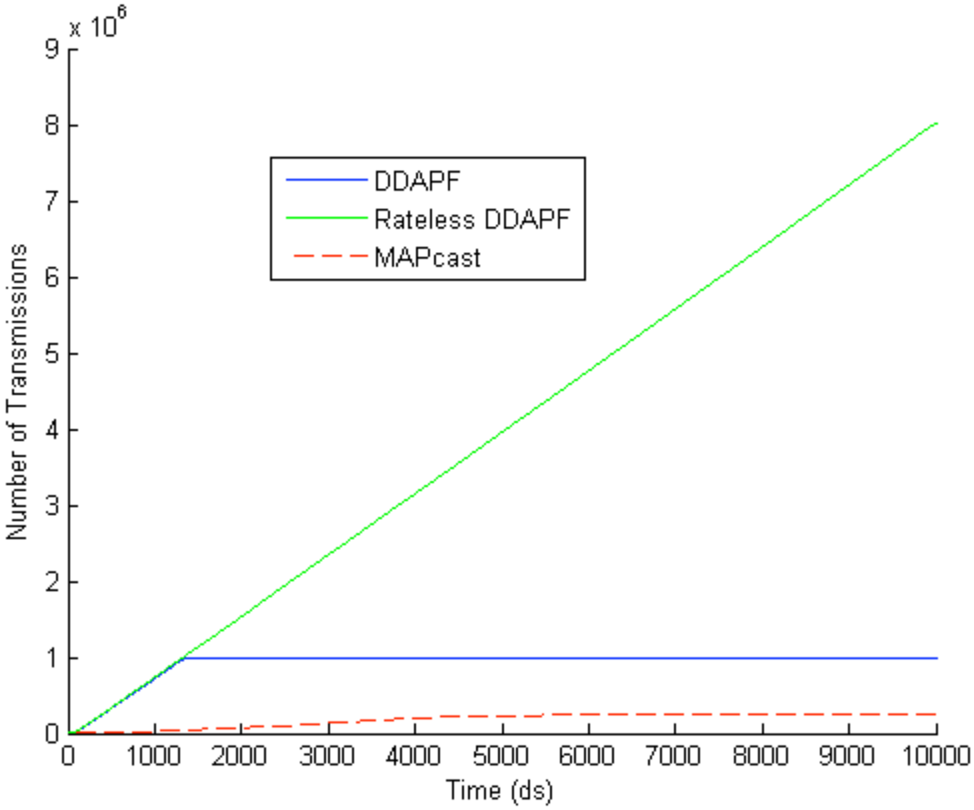

Figure 17b. Power Consumption - MAPcast vs. DDAPF with Rateless Coding

Fig. 18a and 18b show the reachability and power consumption, respectively, for a probabilistic forwarding model using rateless coding at the source at well as decoding and re-encoding at nodes that are able to decode the entire message. This is similar to the Collaborative rateless broadcast (CRBcast) protocol in [11,12], but without the additional packet requests and advertisements associated with the forwarding node recoding.

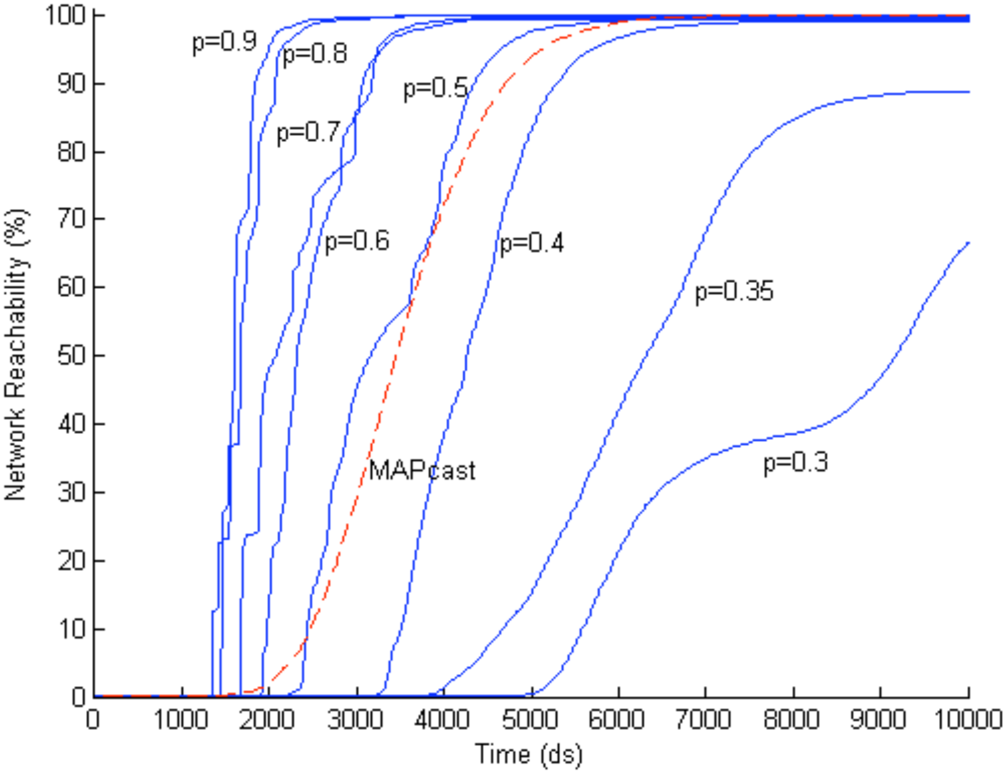

Figure 18a. Reachability - MAPcast vs. CRBcast with forwarding probability *p*.

These results show that while the implemented CRBcast with forwarding probability *p≥0.4* can deliver reliabilities on par with MAPcast, MAPcast achieves large power savings (Fig. 18b). The power consumption of MAPcast is only 20% of that of the next best method (i.e. CRBcast with forward probability *p* = 0.4). This gain is mainly achieved because CRBcast was developed for stationary networks while MAPcast is designed taking into account the mobility of nodes in a MANET.

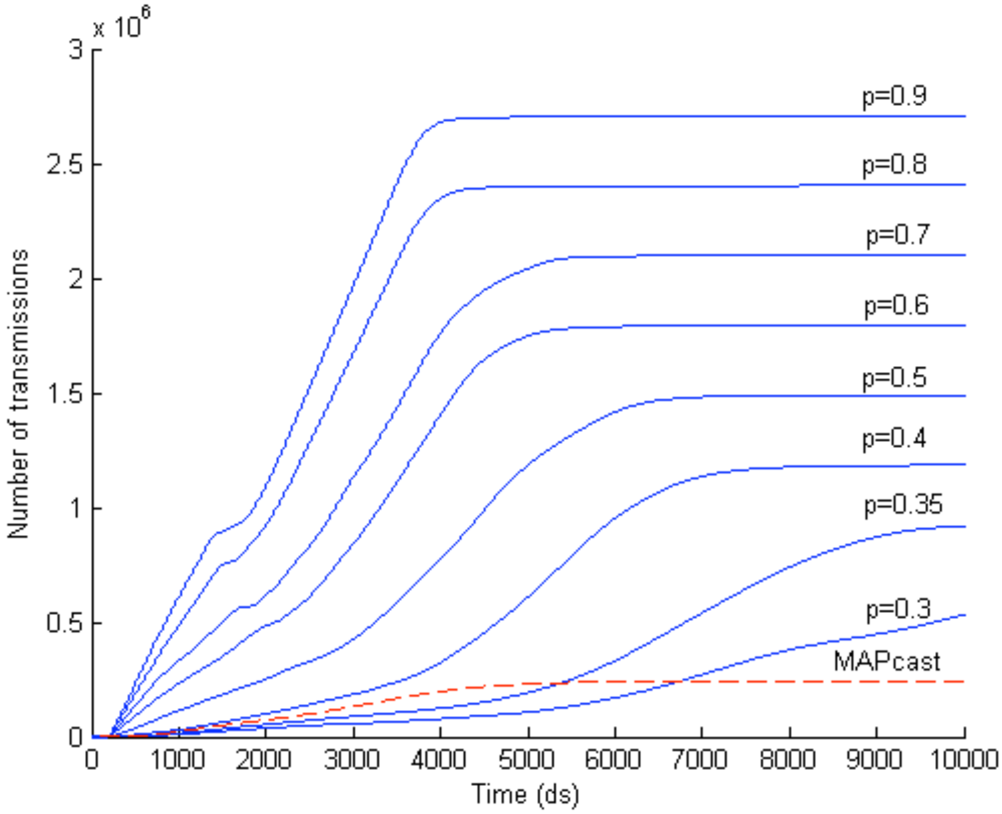

Figure 18b. Reachability - MAPcast vs. CRBcast with forwarding probability *p*.

CHAPTER V

CONCLUSION

A. Remarks

In this thesis, we considered the problem of reliable and energy efficient broadcasting in mobile ad-hoc networks. We proposed a broadcasting protocol, referred to as MAPcast, which is based on adaptive probabilistic broadcasting and rateless coding. We have seen how MAPcast can achieve great power savings over other probabilistic broadcast methods. We have also shown the large gain in reliability offered by rateless coding at the source as well as relaying nodes. The scheme that has closest result to MAPcast is CRBcast, which we showed MAPcast outperforms it by 80% less energy consumption.

The MAPcast protocol has been shown to be a large improvement over existing protocols in terms of energy efficiency. However this improvement comes at the expense of latency. MAPcast generally has a higher latency than other schemes. It is this extra time that MAPcast exploits to achieve its high gains. MAPcast lets the nodes' mobility carry the burden of data distribution rather than use extra energy in the form of redundant transmissions. By waiting and allowing multiple transmission attempts the node can "choose" better transmission opportunities rather than waste a transmission for mediocre results.

B. Future Work

Additional investigation and expansion of MAPcast is desired. The protocol could be tested against mobility patterns with higher mobility (larger Vmax) as well as less randomness in the movements of the nodes (i.e. set patterns of movement). Additionally we propose to add a feature to MAPcast that would reschedule future transmission attempts to occur immediately if a node is in a region of prime transmission opportunity, based on the number of neighbors that lack the advertised packet. In this method, if a transmission attempt is assigned a forwarding probability that was higher than some threshold (e.g. 90%), then the node in question would change the time scheduled for transmission of all packets in its schedule to the current time. This would effectively cause the node to attempt all of it's transmissions for immediate consecutive time slots. Alternatively the stimulus for this action could be based on the number of ACK returns, rather than the ratio of ACK/NACK which the probability is set from. In this scenario a node would reschedule its transmissions if the node receives some threshold of ACK messages (e.g. 10) in response to an ADV message. These changes could help the MAPcast protocol to further exploit the mobility pattern, by taking advantage of these high efficiency (in terms of nodes reached per transmission) situations.

REFERENCES

[1] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath, "Flooding for reliable multicast in multi-hop ad hoc networks," *Proc. Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication*, pp. 64-71, 1999.

[2] S.Y. Ni, Y.C. Tseng, Y.S. Chen and J.P. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pp 151-162, 1999.

[3] Y.C. Tseng, S.Y. Ni, and E.Y. Shih, "Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network," *IEEE Transactions on Computers*, vol. 52, no. 5, May 2003.

[4] J.S. Kim, Q. Zhang, and D.P. Agrawal, "Probabilistic broadcasting based on coverage area and neighbor confirmation in mobile ad hoc networks," *IEEE Comm. Society Globecom 2004 Workshops*, pp. 96-101, 2004.

[5] Q. Huang, Y. Bai, and L. Chen, "Efficient lightweight broadcasting protocols for multi-hop ad hoc networks," *The 17th Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, 2006.

[6] B. Williams, and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*. pp. 194-205, 2002.

[7] B.D. Walker, J.K. Glenn, and T.C. Clancy, " Analysis of simple counting protocols for delay-tolerant networks," *Proceedings of the second ACM workshop on Challenged networks*, pp. 19-26, 2007.

[8] P. Kyasanur, R.R. Choudhury, and I. Gupta, "Smart gossip: an adaptive gossip-based broadcasting service for sensor networks," *The Third IEEE International Conference on Mobile Ad-hoc and sensor Systems (MASS 2006)*, Vancouver, Canada. October 2006.

[9] N. Li, J.C. Hou, "A scalable, power efficient broadcast algorithm for wireless networks," *Wireless Networks*, vol. 12, no. 4., pp. 495-509, August 2006.

[10] B. Carbunar, A. Grama, and J. Vitek, " Redundancy and coverage detection in sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 1, pp. 94-128, February 2006.

[11] N. Rahanavard, B. N. Vellambi, F. Fekri, "CRBcast: a reliable and energy-efficient broadcast scheme for wireless sensor networks using rateless codes," *IEEE Transactions on Wireless Communications*, vol. 7, no. 12, part 2, pp. 5390-5400, December 2008.

[12] N. Rahnavard and F. Fekri, "CRBcast: A collaborative rateless scheme for reliable and energy-efficient broadcasting in wireless sensor networks," in Proc. of *ACM/IEEE 5th International conference on Information Processing in Sensor Networks (IPSN'06)*. pp. 276-283. Nashville, TN, April 2006.

[13] I.A. Khan, A. Javaid, and H.L. Qian, "Coverage-based dynamically adjusted probabilistic forwarding for wireless mobile ad hoc networks," *Proceeding of the 1st ACM international workshop on Heterogeneous sensor and actor networks*, pp. 81-88, May 2008.

[14] I.A. Khan, A. Javaid, and H.L. Qian, "Distance-based dynamically adjusted probabilistic forwarding for wireless mobile ad hoc networks," *Wireless and Optical*

Communications Networks, 2008. WOCN '08. 5th IFIP International Conference on, May 2008.

[15] J. Wu, and F. Dai, “Generic distributed broadcast scheme in ad hoc wireless networks,” *IEEE Transactions on Computers*, vol. 53, no. 10, pp. 1343-1354, October 2004.

[16] F. Ye, G. Zhong, S. Lu, and L. Zhang, “Gradient broadcast: a robust data delivery protocol for large sensor networks,” *Wireless Networks*, vol. 11, no. 3, pp. 285-298, May 2005.

[17] l. Rodrigues, S. Handurukande, J. Pereira, R. Guerraoui, A.M. Kermarrec, “Adaptive gossip-based broadcast,” *Dependable Systems and Networks, 2003. Proceedings. 2003 International Conference on*, pp. 47-56, June 2003.

[18] M. Luby, “LT codes,” *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pp. 271-280, 2002.

[19] A. Shokrollahi, “Raptor Codes,” *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp 2551-2567, June 2006.

[20] A. Venkiah, P. Piantanida, C. Poulliat, P. Duhamel, and D. Declercq, “Rateless coding for quasi-static fading channels using channel estimation accuracy,” *IEEE International Symposium on Information Theory*, 2008.

[21] C. Wu and B. Li, “Outburst: efficient overlay content distribution with rateless codes,” *Lecture Notes in Computer Science*, vol. 4479/2007, pp. 1208-1216, 2007.

[22] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. “Energy-efficient broadcast and multicast trees in wireless networks.” *Mobile Networks and Applications*. Vol. 7, No. 6. pp 481-492. Springer Netherlands. December, 2002.

[23] B.N. Vellambi, R. Subramanian, F. Fekri, M. Ammar, “Reliable and efficient message delivery in delay tolerant networks using rateless codes,” *Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking*, pp. 91-98, 2007.

[24] G. Karlsson, V. Lenders, and M. May, “Delay-tolerant broadcasting,” *IEEE Transactions on Broadcasting*, vol. 53, no. 1, March 2007.

VITA

Jason Douglas Pitts

Candidate for the Degree of

Master of Science

Thesis: EFFICIENT BROADCAST IN MOBILE ADHOC NETWORKS UTILIZING RATELESS CODING

Major Field: Electrical Engineering

Biographical:

Personal Data: Jason Pitts was born in Tulsa, OK, to Douglas and Viola Pitts. He was raised in Verdigris, OK where he attend K-12 at Verdigris Public Schools. Jason graduated High School in the Spring of 2003. He went on to attend classes at Oklahoma State University. Jason enjoys reading, gaming, hiking and cooking. Jason is also very interested in coding and encryption and spends time creating codes and attempting to break existing methods. Jason will marry his long time girlfriend, Susan, in the summer of 2009. He has one sister, Jessica, and a cat, Alice.

Education:
Completed the requirements for the Master of Science in Electrical Engineering at Oklahoma State University, Stillwater, Oklahoma in May, 2009.

Graduated with a Bachelor of Science degree in Electrical Engineering from Oklahoma State University in December, 2007.

Experience:
Teaching assistant for ECEN4013.
Research assistant under Dr. Nazanin Rahnavard.
Research assistant under Dr. Keith Teague and Dr. George Scheets.
Research assistant under Dr. Sohum Sohoni.

Professional Memberships:
Eta Kappa Nu Member

Name: Jason Douglas Pitts Date of Degree: May, 2009

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: EFFICIENT BROADCAST IN MOBILE ADHOC NETWORKS UTILIZING RATELESS CODING

Pages in Study: 54 Candidate for the Degree of Master of Science

Major Field: Electrical Engineering

Scope and Method of Study: This paper presents a novel efficient broadcast protocol for mobile ad hoc networks referred to as mobile adaptive probabilistic broadcast (MAPcast). MAPcast integrates application-layer rateless coding with adaptive probabilistic packet forwarding amongst the nodes of a mobile network without the need for information from nodes other than an immediate neighbor (one hop). MAPcast is developed as a method to reduce the number of overall packet transmissions, thus conserving power, while ensuring very high network reachability. MAPcast is compared with existing methods of data broadcast to examine the energy efficiency and reachability provided using the protocol outlined in this paper.

Findings and Conclusions: Simulation results indicate that MAPcast has superior performance both in terms of network reachability as well as power consumption.

ADVISER’S APPROVAL: Dr. Nazanin Rahnavard