

Analysis of Core-Assisted Routing in Opportunistic Networks

Muhammad Abdulla and Robert Simon

Department of Computer Science

George Mason University

Fairfax, VA 22030

Email: {mabdulla, simon}@cs.gmu.edu

Abstract—Opportunistic Networks (*ONs*) are a newly emerging type of Delay Tolerant Network (*DTN*) systems that opportunistically exploit unplanned contacts among nodes to share information. As with all DTN environments *ONs* experience frequent and large delays, and an end-to-end path from the source to destination may only exist for a brief and unpredictable period of time. Such network conditions present unique challenges to message routing. In this paper, we present the design and performance analysis of a novel core-based routing protocol for *ON* routing. Under the assumption that messages will have a delivery time constraint, we then provide a set of analytical results for rapid modeling and performance evaluation for the basic performance metrics of message delay, message delivery ratio, and buffer occupancy. We have implemented our protocol in *ns-2*, and our simulation results show that our protocol is quite effective and our analysis is accurate.

I. INTRODUCTION

Opportunistic Networks (*ONs*) are a newly emerging type of Delay Tolerant Networked (*DTN*) systems that can *opportunistically* exploit unplanned contacts between nodes to share information. As with all DTN environments *ONs* experience frequent and long lasting partitions [3–5]. In an *ON* an end-to-end path between the source and the destination may only exist for a brief and unpredictable period of time. With the increased use of wireless mobile devices, many new network applications fall into this category, such as wildlife tracking, disaster recovery and emergency response systems, and social-based peer-to-peer networks that rely on human mobility.

Our work focuses on the performance analysis of a *core-based* approach for *ON* routing. A core is a specially designated node within the *ON* that has special responsibilities within the “Store-Carry-Forward” message transfer paradigm. Core nodes simplify network management functions such as multicasting, group membership and network security, by centralizing these activities inside of cores. For instance, non-core nodes can manage their security associations via interactions with core nodes, rather than relying on pure peer-to-peer approaches.

In this paper we present the performance analysis of the core-assisted routing scheme for opportunistic networks. Specifically, using existing mobility models we provide an analytical model for core-assisted routing scheme and derive analytical results for fundamental performance metrics such as message delay, message delivery ratios, and buffer occupancies

when there is a message expiration time associated with messages. In our analysis, we assume that nodes have no prior or on-going knowledge of node connectivity or mobility patterns.

For the analysis of core-assisted routing, we first consider the simple Direct Transmission routing scheme for DTNs. Direct Transmission is a standard DTN technique that requires the source to wait until it is in direct contact with the destination before transmitting the message. Based on the analysis of Direct Transmission scheme, we extend our analysis to our Direct Transmission Plus Core (*DTPC*) routing scheme. Such analytical results can be used for rapidly estimating basic performance parameters for *ON* routing using either a pure Direct Transmission approach or *DTPC*.

As a basic application requirement, message delay has been the focus of much work for DTNs [7, 12, 13]. In this work, besides providing analytical results for the message delay, we extend our analysis to other important performance metrics such as the message delivery ratio and buffer occupancy. For practical applications, we incorporate time constraints such as message expiration times in our analysis of abovementioned performance metrics. For this study we have implemented *DTPC* in *ns-2*, and will show the performance and tradeoffs of our method.

The rest of the paper is organized as follows. Section II goes over some related work. Section III discusses *DTPC* and section IV provides an analytical model for performance analysis of our scheme. Section V describes our experimental results. Finally, Section VI gives some conclusions and directions for future work.

II. RELATED WORK

Opportunistic networks are a type of Delay Tolerant Networking (*DTN*), so much relevant related work is motivated by research performed for DTNs. Because of frequent network partitions in the DTN environment many traditional routing techniques for Mobile Ad Hoc Networks will not work properly [1, 5, 10]. This fact has led to recent interest in developing new approaches for routing in a DTN environment. The basic routing paradigm for effective routing in DTNs is to use the *Store-Carry-Forward* approach, where intermediate nodes keep the messages until new links come up in the path to the destination.

One general class of proposed DTN routing algorithms assumes some level of knowledge regarding node mobility and connectivity. For instance, Jain et. al. formulates the DTN routing in terms of a directed multi-graph, where more than one edge may exist between a pair of nodes [10]. Such multiple edges exist because there may be more than one distinct physical connections or different network links may only be available at different time intervals. By using different levels of information regarding connectivity and/or mobility, routing decisions can be made at individual nodes.

Although the knowledge about node connectivity is useful for making routing decisions, such information may not be available to the nodes in the network, especially in Opportunistic Networks where contacts are unpredictable. Under such conditions different routing approaches are necessary for effective message delivery.

Recent DTN routing approaches concentrate on trading off message complexity versus increasing the likelihood of message delivery. To limit the number of messages *single copy* routing schemes allow only one copy of the message at a time to be present in the network [13]. *Direct Transmission* is the simplest form of single copy routing, where each source node keeps its messages until it comes into direct contact with the respective destination nodes. Under this scheme only one message transfer is made per delivered message, incurring minimal message passing. However, in intermittently connected networks, such an approach may produce low delivery ratios and has an unbounded delivery delay [8].

One way to improve the performance of a single copy approach is to have multiple copies of the same message within the network. One policy to implement a multi-copy scheme is to use flooding. One example is Epidemic Routing [17]. In Epidemic Routing when a pair of nodes comes into contact the nodes exchange any missing packets. Given enough storage space and bandwidth, Epidemic routing can be used to reliably disseminate data across the network. However, due to its large overhead, a flooding scheme such as Epidemic Routing may not be applicable under circumstances where storage and power supplies are limited.

To address overhead problems caused by flooding, different forms of controlled flooding have been proposed, including message expiration times, limiting the number of hops a message can travel, and using active and passive “curing” techniques [9, 12]. Controlling the number of copies spread for a message is also an effective approach for controlled flooding, for which Spyropoulos et al present Spray and Wait [14]. In this method, a total of L copies of a message are initially spread to other “relay” nodes. If the destination is not found in this phase, each of the nodes carrying a copy of the message will perform direct transmission. No mobility or connectivity information regarding the nodes in the network are assumed to be known for this scheme to work. Although *DTPC* also uses a limited number of nodes to carry messages only a set of designated nodes exchange and carry messages in the network.

In the *Data MULEs* approach proposed in [11] a number of mobile nodes perform random walks to collect packets, buffer

them, and deliver them to wired access points. The sensor nodes are static. For *DTPC* we assume that all the nodes can be mobile. A message can be delivered to the destination either through a core, or by the source itself. Further, the core nodes can use flexible message exchange policies amongst themselves.

In [20] a route planning strategy is introduced using message ferries that travel on a trajectory to provide communication services. Either the message ferries choose a trajectory to contact nodes, or the nodes can move near to pre-defined trajectory at a certain time to exchange packets. This scheme requires either prior knowledge or online collaboration regarding node mobility information. Tariq et. al. [16] discuss route design issues for a single ferry based on node mobility characteristics, without requiring online collaboration between the node and the ferry. *DTPC* differs from message ferrying in that the movement of core nodes is not assumed to be controlled.

Recent work in Opportunistic Routing environments include [3], which tries to balance message replication with erasure coding, and [4], which deals with content distribution. In contrast *DTPC* focuses directly on developing a high-performing message transfer protocol.

Recently there has been considerable interest in developing analytical models for the performance of DTN routing schemes. [7, 12–14, 19]. Much of the work focuses on message delay. However, in many cases, we are also interested other elements such as message delivery ratio (MDR) and buffer occupancies. Further, time constraints such as message expiration times need to be addressed in the performance analysis, because such constraints either occur as an application level requirement, or as a routing policy [9, 12].

In this work, we present an analytical model for the performance of *DTPC*. Our starting point is to assume that nodes come into contact with other nodes according to the standard exponential distributions used in many synthetic mobility models. One of the contributions of our analysis is that we present results for the three basic metrics of message delay, delivery ratio, and buffer occupancy in the presence of time constraints.

III. THE *DTPC* PROTOCOL

This section describes our core-based protocol for *ON* routing, called *Direct Transmission Plus Core*, or *DTPC* for short. As the name implies, we use two types of transfer policies, either direct delivery between a source node and its destination, or a message transfer between a source node and a core node. As noted in earlier sections a pure Direct Transmission strategy simply means that after generating a message the source waits until it comes into direct contact with the destination [13]. The main advantage of this scheme is that it incurs minimum data transfers for message deliveries. It has been shown that the message delay in direct transmission routing scheme gives the worst-case performance bound for non-adversarial routing schemes for DTNs [15]. We also introduce two types of source to core transfer policies. We will

discuss the core transfer policies after a general description of *DTPC*.

A. Mechanisms

We assume that nodes are synchronized in time on the order of seconds. The purpose of time synchronization is to allow core nodes to delete messages that are expired. If this is not possible then a standard Time-To-Live mechanism using a countdown timer can be used.

We also assume as part of its basic data link protocol that each node possesses the capability of determining when it is in range of another node. This portion of *DTPC* is called *DL-Hello* (for Data-Link Hello) protocol. *DL-Hello* is easily by simply adding a module in any wireless datalink protocol to periodically send out a special *DL-Hello* message. This message contains a node's identifier, a flag set to indicate whether or not the node is a core, and a transaction flag, which is described below.

Nodes are distinguished as either core nodes or non-core nodes. As explained in the previous sections, the purpose of the core nodes is to reduce the number of message transmissions and buffering requirements, to simplify network management activities such as security, and to provide the building blocks for core-based multicast schemes, etc. However, for this work we only focus on the the rules and analysis for message transfer.

For *DTPC* we assume that all nodes want to unicast messages to other nodes. This is represented as follows: At time t node i produces a message for a destination dst . The format of this message is

$$msg^i(dst, sn, t, T_{sn}, payload)$$

where sn is a sequence number used for duplicate detection, t is the generation time, T_{sn} is the message timeout value, and $payload$ contains the actual data. The timeout value is interpreted as meaning that this message is set to expire at time $t + T_{sn}$. The timeout can be used for memory management and as a means for an application to signify data freshness. All nodes that generate messages are called source nodes (both core and non-core).

Each source node i keeps a list of un-delivered and non-expired $msgs^i$ in a local buffer. When a message times-out a source node simply deletes the message from its buffer. *DL-Hello* signifies to the *DTPC* layer that a node is in range node i and delivers to node i the other nodes' identifier, whether or not the other node is a core node, and the value of the transaction flag. At that point message summaries may be exchanged according to the message transfer procedure described below.

One policy issue that must be addressed is what happens to the messages in node i 's queue if the newly contacted node is a core. We have defined two types of non-core to core copy policies. The first is called *Copy-to-Core*. In this case node i does not delete its message queue after it transmits its messages. Using *Copy-to-Core* a message can be delivered to the destination either by the source node or by a core.

The second policy is called *Dump-to-Core*. For *Dump-to-Core* node i deletes all its buffered messages after it transmits its messages to the core.

B. Message Transfer Procedure

The message transfer procedure is invoked by each node i when the *DL-Hello* layer reports that an arbitrary node j has come into range. The *DL-Hello* layer also reports whether node j is a core node or a node core node, along with the value of the transaction flag. There are three cases to consider:

- Node i and Node j are both non-core nodes. In this case both nodes examine their queues and determine if they have any messages that are pending for the other node. They then simply exchange the relevant messages, or send to the other node that they have no pending message. After a transaction acknowledgment both nodes then delete any exchange messages.
- Node i is a non-core node and node j is a core node. In this case node i will send all its messages to node j . Node j uses the sequence number for each message for duplicate detection. Node j acknowledges this transaction, at which point for the *Dump-to-Core* policy node i deletes all its messages. Node j then sends any pending messages it has to node i .
- Both node i and node j are core nodes. In this case both nodes exchange message summary lists sorted by dst and sn . Each node then produces a request list for the messages it does not have. Based on the request list exchange both nodes will obtain any missing messages, and both sides acknowledge a successful transaction.

During this procedure it is possible that other nodes will come into contact range to either node i or node j . We include a transaction flag as part of the *DL-Hello* protocol. This flag is set when message transfers are occurring, as a way of preventing a new transfers from taking place. Although more efficient mechanisms are possible that allow concurrent transfers we have found this to be the simple and effective way of streamlining the transfer procedure and not require more complex mechanisms due to race conditions. The transaction flag allows a node to continue to poll the other side to determine when a transaction is finished.

IV. ANALYSIS OF CORE-ASSISTED ROUTING SCHEME

In this section, we analyze fundamental performance metrics for Direct Transmission and core-assisted routing schemes, including *Message Delivery Ratio* (MDR), *Delay of Delivered Messages*, and *Buffer Occupancy*. Our starting point is to base our analysis on the commonly used synthetic mobility models of Random Waypoint and Random Direction. In particular, we assume that the inter-arrival time between successive contacts is exponentially distributed. This assumption is supported by the results presented in [7], which shows that nodal inter-meeting times are nearly exponentially distributed when transmission ranges are small compared to the network area size. This should normally be the case for opportunistic networks. Using empirical observation suggests that Markovian models

for routing schemes for such networks can lead to accurate performance predictions. Exponentiality of inter-meeting times is also assumed in other studies for DTN scenarios [14, 16].

In our analysis, we use γ to denote the rate of inter-meeting times, and use λ for message generation rate of each node. Messages are assumed to have a message expiration time, T_x , beyond which they will be dropped. For simplicity we assume that the time taken to execute the *DL-Hello* protocol is zero, and that the transaction flag is always unset.

A. Analysis of Direct Transmission Routing Scheme

Direct Transmission is the simplest routing scheme, where the sender waits till it comes into contact with the destination to deliver a message. It has the upper bound for message delay for any non-adversarial mobility-assisted routing scheme [15]. Although not applicable for many scenarios due to large delays and low delivery ratios, Direct Transmission can be used as the basis for performance analysis of other routing approaches, including our core-assisted routing scheme.

1) *Message Delivery Ratio*: Provided that node arrival times are exponentially distributed with a rate of γ , for a message entering in the queue at time 0 the probability that the message is delivered before it is expired can be given in the form of CDF as follows:

$$E[R] = 1 - e^{-\gamma T_x} \quad (1)$$

where T_x is the message expiration time. Here we assume that no messages are dropped due to buffer overflow.

2) *Delay of Delivered Messages*: Given message expiration time T_x , messages get delivered if the destination is reached within T_x , or it will be dropped. From an application's point of view, we are only interested in the expected time that the delivered messages spend in the buffer queue before it gets delivered, i.e., the delay of delivered messages, which is given by the following theorem.

Theorem 1: Given that the node meeting times are exponentially distributed with a rate of γ , and the message expiration time is T_x , then the expected message delay of delivered messages, ED_{dt}^x , under the Direct Transmission scheme is given as:

$$ED_{dt}^x = \frac{1}{\gamma} - \frac{e^{-\gamma T_x}}{1 - e^{-\gamma T_x}} T_x$$

Proof:

Given that the inter-arrival times of nodes are exponentially distributed with a rate of γ , the probability of a message being delivered to the destination at time t after it enters the queue can be given by

$$f(t) = \gamma e^{-\gamma t}$$

For delivered messages, the probability function given above becomes a conditional probability for the messages that are delivered:

$$f_d(t) = \frac{f(t)}{P(t \leq T_x)} = \frac{\gamma e^{-\gamma t}}{1 - e^{-\gamma T_x}} \quad (2)$$

where $P(t \leq T_x)$ denotes the probability that the destination is reached before T_x , which is given by the CDF of $f(t)$.

Therefore, the expected waiting time, $E_d(T)$, of a delivered message can be written as

$$\begin{aligned} E_d(T) &= \int_0^{T_x} t f_d(t) dt \\ &= \frac{\gamma}{1 - e^{-\gamma T_x}} \int_0^{T_x} t e^{-\gamma t} dt \\ &= \frac{\gamma}{1 - e^{-\gamma T_x}} \left[\frac{-t e^{-\gamma t}}{\gamma} \Big|_0^{T_x} + \frac{1}{\gamma} \int_0^{T_x} e^{-\gamma t} dt \right] \\ &= \frac{1}{\gamma} - \frac{e^{-\gamma T_x}}{1 - e^{-\gamma T_x}} T_x \end{aligned} \quad (3)$$

This result gives us the expected delay of a message with expiration time T_x when the arrival rate of the destination is γ . Since relation given in (3) will be used later, we define $\Phi(\gamma, t_x)$ as a function a of arrival rate and message expiration time as follows:

$$\Phi(\gamma, t_x) = \frac{1}{\gamma} - \frac{e^{-\gamma t_x}}{1 - e^{-\gamma t_x}} t_x \quad (4)$$

It can be shown that the value of $\Phi(\gamma, t_x)$ is upper bounded by $\min\{t_x, 1/\gamma\}$, and approaches $1/\gamma$ when $t_x \rightarrow \infty$.

3) *Buffer Occupancy*: Given the message generation rate, λ , and inter-arrival (meeting) rate of nodes, γ , we can find the number of nodes in the buffer using a queueing system model with vacations, where the message arrival rate is λ and vacation time is distributed exponentially with a rate of γ . Since the message transfer time is very small compared to arrival times, we can take the service rate as infinite for simplification. We assume that no messages are dropped due to buffer overflow.

First, we find the expected time that a message spends in the buffer, whether it is delivered or dropped due to message expiration. We already obtained the expected delay of delivered messages, $E_d(T)$, in *Theorem 1* above. The time, $E_x(T)$, that an expired message spends in buffer is simply T_x . Therefore, the expected time, $E(T)$, that a message spends in the buffer is given as follows:

$$\begin{aligned} E(T) &= (1 - e^{-\gamma T_x}) E_d(T) + e^{-\gamma T_x} E_x(T) \\ &= \frac{1 - e^{-\gamma T_x}}{\gamma} - T_x e^{-\gamma T_x} + T_x e^{-\gamma T_x} \\ &= \frac{1 - e^{-\gamma T_x}}{\gamma} \end{aligned} \quad (5)$$

Given message generation rate of λ , we can give the expected number of messages in the buffer, \bar{N} , as follows by using Little's Law:

$$\begin{aligned} \bar{N} &= \lambda * E(T) \\ &= \frac{\lambda}{\gamma} (1 - e^{-\gamma T_x}) \end{aligned}$$

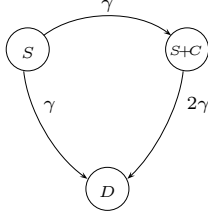


Fig. 1. State Transition Diagram for Single-core Routing Scheme

B. Analysis of Single-core Routing Schemes

In our analysis of core-assisted routing, we consider the Copy-to-Core scheme discussed earlier. In this scheme, when a non-core node N meets a core node C , N delegates all of its undelivered packets to C that it has not previously delegated to C .

Below, we analyze performance metrics under core-assisted routing scheme where there is only one core node in the system. The state transition diagram for single-core routing scheme is shown in Figure 1. In this diagram, the state S denotes the scenario when only the source node has the copy of a message, state D denotes that the message has been delivered to the destination, and state $S+C$ denotes that both the source and the core have copies of the message.

According to the properties of exponential distribution, the rate at which the source meets either the destination or the core node first is 2γ . The probability of either of these two event happening is equal to 0.5. Further, if the message is delivered to the core node first, the rate at which the destination meets either the source or the core is also 2γ . We use $\phi = 2\gamma$ below for simplification.

1) *Message Delivery Ratio*: Similar to the analysis above, the message delivery ratio, MDR_D , under the condition that the message is delivered directly to the destination by the source is given as, following (1),

$$MDR_D = 1 - e^{-\phi T_x}$$

The message delivery ratio, MDR_C , under the condition that the message is delivered after it is delegated to the core is given as

$$\begin{aligned}
 MDR_C &= \int_0^{T_x} f_d(t)P_r(t)dt \\
 &= \int_0^{T_x} \phi e^{-\phi t}(1 - e^{-\phi(T_x-t)})dt \\
 &= \int_0^{T_x} \phi e^{-\phi t} - \phi e^{-\phi T_x} dt \\
 &= 1 - e^{-\phi T_x} - \phi T_x e^{-\phi T_x}
 \end{aligned} \tag{6}$$

Here $f_d(t)$ is the PDF denoting the probability of destination meeting the core at time t , and is given by $f_d(t) = \phi e^{-\phi t}$. The probability of delivery, $P_r(t)$, if the core is met at time t is given as $P_r(t) = (1 - e^{-\phi(T_x-t)})$.

Combining the results, the expected MDR, MDR_{c1} , under the 1-core scheme is

$$MDR_{c1} = \frac{MDR_D + MDR_C}{2} = 1 - e^{-\phi T_x} - \frac{\phi T_x e^{-\phi T_x}}{2}$$

2) *Message Delay*: Following (3), the expected delay, T_D , of a message directly delivered to the destination is given as

$$T_D = \Phi(\phi, T_x) = \frac{1}{\phi} - \frac{e^{-\phi T_x}}{1 - e^{-\phi T_x}} T_x$$

The expected delay, T_C , of a message delivered after it is delegated to the core is given as

$$T_C = \int_0^{T_x} \frac{f_d(t)P_r(t)ED(t)}{P_D} dt$$

Here, $f_d(t)$ denotes the probability that the source node meets the core at time t . Under this event, $P_r(t)$ gives the probability that the message will be delivered and $ED(t)$ denotes the expected delay of the message. Since we only consider the delivered messages, we use the probability of delivery, P_D , in the conditional probability given above.

The probability of the message reaching the destination, either by the source or by the core, after the message is delegated at time t is given by

$$P_r(t) = 1 - e^{-\phi(T_x-t)}$$

The expected delay, $ED(t)$, of the message is given as

$$\begin{aligned}
 ED(t) &= t + \int_0^{T_x-t} z \phi e^{-\phi z} dz \\
 &= t + \frac{1 - e^{-\phi(T_x-t)} - \phi(T_x-t)e^{-\phi(T_x-t)}}{\phi}
 \end{aligned}$$

Since we only are only considering delay of delivered messages, the probability of message delivery, P_D , is equal to the message delivery ratio given in Equation (6):

$$P_D = MDR_C = 1 - e^{-\phi T_x} - \phi T_x e^{-\phi T_x}$$

Taken together, we have

$$\begin{aligned}
 T_C &= \int_0^{T_x} \frac{f_d(t)P_r(t)ED(t)}{P_D} dt \\
 &= \int_0^{T_x} \frac{\phi e^{-\phi t}(1 - e^{-\phi(T_x-t)})ED(t)}{1 - e^{-\phi T_x} - \phi T_x e^{-\phi T_x}} dt \\
 &= \frac{2 - 3\phi T_x e^{-\phi T_x} - 2e^{-2\phi T_x} - \phi T_x e^{-2\phi T_x} - \phi^2 T_x^2 e^{-\phi T_x}}{\phi(1 - e^{-\phi T_x} - \phi T_x e^{-\phi T_x})}
 \end{aligned}$$

Since we assume that node movements are i.i.d, the probability that the source node meets the destination or the core node first is equal. Therefore, the expected message delay when there is a single core, ED_{c1} , is given by

$$ED_{c1} = \frac{T_D + T_C}{2} \tag{7}$$

3) *Buffer Occupancy*: Upon contacting the core node, a non-core node gives a copy of each message that have not been delegated to core. The non-core node continues to keep a delegated message until the destination is reached or the message is expired. Therefore, the expected buffer occupancy at a non-core node is the same as under the Direct Transmission scheme, as given in (6).

To find the buffer occupancy at the core node, we first find the expected time, ET_c , that a message spends at the core node before it is delivered or expired:

$$\begin{aligned}
ET_c &= \int_0^{T_x} f_d(t)ED(t)dt \\
&= \int_0^{T_x} \phi e^{-\phi t} \frac{1 - e^{-\gamma(T_x-t)}}{\gamma} dt \\
&= \int_0^{T_x} 2\gamma e^{-2\gamma t} \frac{1 - e^{-\gamma(T_x-t)}}{\gamma} dt \\
&= 2 \int_0^{T_x} e^{-2\gamma t} dt - 2e^{-\gamma T_x} \int_0^{T_x} e^{-\gamma t} dt \\
&= \frac{1 - e^{-2\gamma T_x}}{\gamma} - \frac{2e^{-\gamma T_x}(1 - e^{-\gamma T_x})}{\gamma} \\
&= \frac{(1 - e^{-\gamma T_x})^2}{\gamma} \tag{8}
\end{aligned}$$

Here, $ED(t)$ is the expected time that a message spends at the buffer queue of the core node if it is delegated to the core at time t . It is given as $ED(t) = (1 - e^{-\gamma(T_x-t)})/\gamma$, following Equation (5).

Assuming the number of non-core nodes in the system is N , the average number of messages in the buffer of the core, EB_{c1} , can be given as follows according to Little's Law:

$$\begin{aligned}
EB_{c1} &= \frac{N\lambda}{2} * ET_c \\
&= \frac{N\lambda}{2\gamma} (1 - e^{-\gamma T_x})^2 \tag{9}
\end{aligned}$$

C. Analysis of Multi-core Routing Schemes

Since deriving exact analytical results for multi-core routing schemes under time constraints is difficult, in this work we give heuristic approximations for performance metrics based on the state transition diagram depicted in Figure 2. In this figure, special state D denotes that the message is delivered to the destination. Other states denote the number of message copies for a specific message.

Based on the model presented in Figure 2 and the result of simulation studies, we use the following set of recursive relations to obtain reasonable approximations:

$$\begin{aligned}
EMDR^x(i, t_x) &= \frac{i}{i(C+2-i)} F((N-1)\gamma, t_x) \\
&\quad + \frac{i(C+1-i)}{i(C+2-i)} EMDR^x(i+1, t_x - ED_i^x), \quad i \in [1, C] \\
EMDR^x(i, t_x) &= F((C+1)\gamma, t_x), \quad i = C+1 \tag{10}
\end{aligned}$$

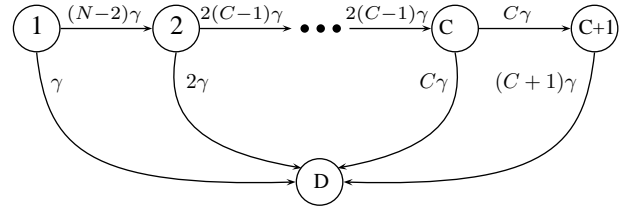


Fig. 2. State Transition Diagram for Multi-core Routing Scheme

where $F(\gamma, t_x) = 1 - e^{-0.7*\gamma t_x}$, and $ED_i^x = \Phi(i(C+2-i)\gamma, t_x)$.

We obtain these relations by considering the probabilities with which the system moves to the next non- D state or to state D , based on the rate at which the system leaves the current state to either of the two states. Expressions for the delay of delivered messages and buffer occupancies can be obtained in similar manner, but with different levels of complexity. Due to lack of space, we omit these heuristic results for delay and buffer occupancies.

V. EXPERIMENTAL RESULTS

In this section, we present our simulation study for the performance analysis in the previous section for core-assisted routing approaches under different settings. Our results were obtained from simulation experiments using *ns-2* and our own code. The goals of the experiments are to validate our analytical model, to show the effectiveness of the core-assisted routing scheme, and to provide insight regarding the changes in performance when the number of nodes, number of cores, and message expiration times are varied.

A. Metrics and Methodology

All of our experiments use the standard *ns-2* mobile wireless models, including the default transmission model which has a 250m radio range. We collected statistics for the average message delivery rate, the average message delivery delay, and the buffer occupancy.

The *average message delivery ratio* is the ratio of delivered messages to the number of messages that should have been delivered to destination nodes. The average message delivery ratio reflects the overall efficiency of the method in delivering messages.

The *average delivery delay* is the average delay of all the messages delivered to destinations. The delay of a delivered message is calculated by subtracting the delivery time by the message generation time.

We also use *buffer occupancy* as a metric to evaluate the buffer requirements of a specific routing method. For the routing schemes that we consider, buffer occupancy can also be used as an indirect estimate of the number of message transfers under the assumption that the protocol overhead is small.

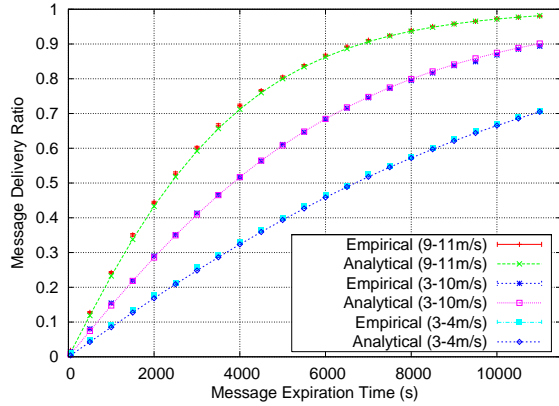


Fig. 3. Message Delivery Ratio in Single-core Routing Schemes Under Varying Speeds

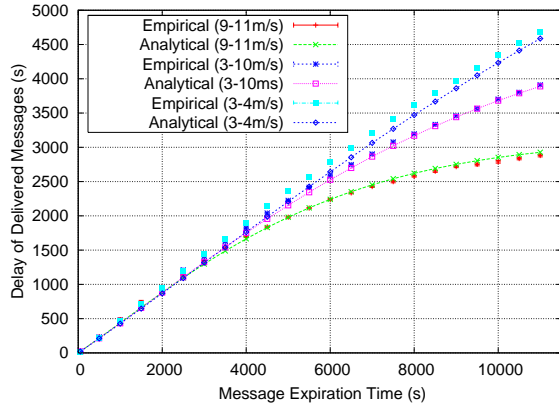


Fig. 4. Message Delay in Single-core Routing Scheme Under Varying Speeds

B. Experimental Settings

The default settings in our simulations are as follows. Each simulation run has 40 nodes in a $6000m \times 6000m$ area. Each node generates a message to another node at random at every 250 seconds on average. Nodes announce their presence using periodic *DL-Hello* messages in every three seconds.

The simulations use the random-waypoint (RWP) mobility model. In RWP nodes randomly choose a point in the area and moves towards that destination with an average speed uniformly distributed between v_{min} and v_{max} . In our simulations, default value for v_{min} is 9, and v_{max} is 11. A large default v_{min} is chosen to provide a steady state faster [18], although we also experiment with lower values. Pause time after reaching the destination point is 3 seconds. All the experiments are run 28 times with random seeds, and the data points are plotted with 95% confidence intervals.

C. Experimental Results

Figure 3 shows the change in message delivery ratio as the speed of nodes and the message expiration times change. We can observe the the analytical values, which are obtained using Equation (6), closely agree with the experimental values.

Figure 4 shows the empirical and analytical values for the delay of delivered message when the node speed and message

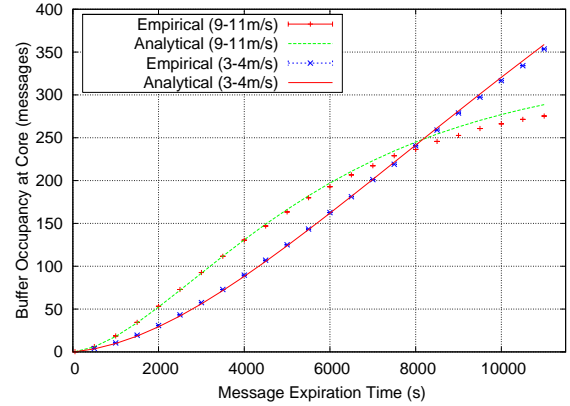


Fig. 5. Buffer Occupancy in Single-core Routing Schemes Under Varying Speeds

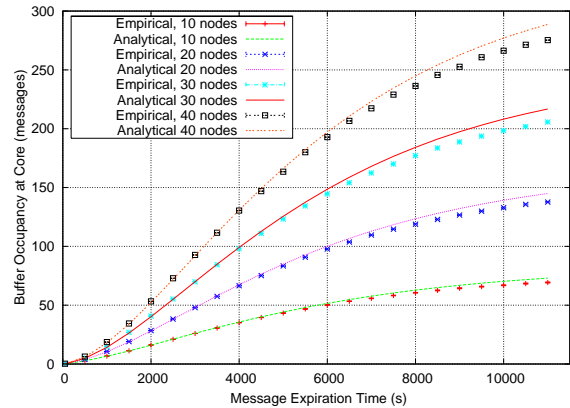


Fig. 6. Buffer Occupancy in Single-core Routing Schemes Under Varying Number of Nodes

expiration times are changed. As shown in [7], the increase in the speed of node movement increase node inter-meeting rate. With the increase in inter-arrival rate, expected message delay decreases, gradually approaching an upper bound that can be obtained when $T_x \rightarrow \infty$. Analytical results for the delay are obtained using Equation (7).

Changes in buffer occupancy when the message expiration time is varied are shown in Figure 5 for two different scenarios with different node inter-arrival rates. As shown in the figure, when message expiration times are small, the buffer occupancy at the core is lower when the inter-arrival rate is low. This is because messages expire before they can be transferred to the core node, confirming lower message delivery ratios as shown above. However, when message expiration times are large, the buffer occupancy for low inter-arrival rate case is larger than that of higher inter-arrival rate. This is because the messages stay longer when inter-arrival rate is low, increasing the number of messages in the buffer.

Figure 6 shows the number of messages at the core node when the number of nodes in the system changes. We can see that the buffer occupancy increase as total number of nodes increases, as given in Equation (9), and that analytical values agree with experimental results.

Message delivery ratios when there are one or more core

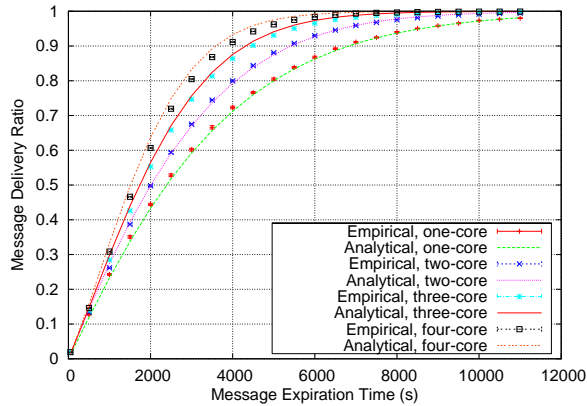


Fig. 7. Message Delivery Ratio in Multi-core Routing Schemes

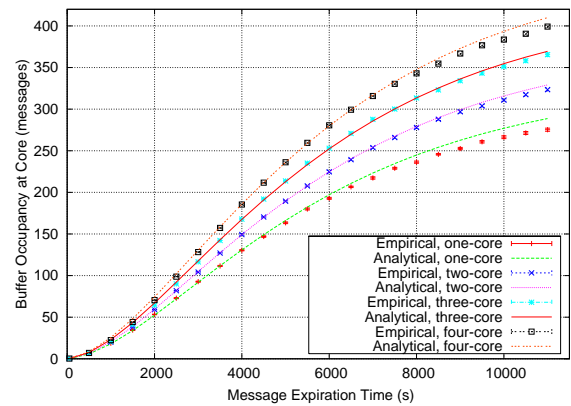


Fig. 9. Buffer Occupancy in Multi-core Routing Schemes

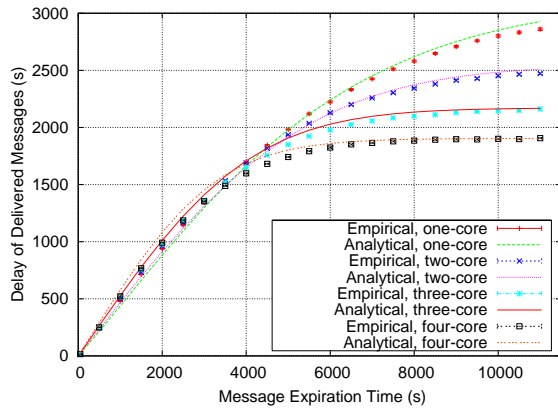


Fig. 8. Message Delay in Multi-core Routing Schemes

nodes in the system are shown in Figure 7. Analytical results are obtained using the heuristic approximations given in previous section. Results for message delay are given in Figure 8. As observed earlier for single-core case, the message delay approaches different upper bounds when the number of cores varies. Figure 9 shows the buffer occupancy at each core. Increasing the number of cores increases buffer occupancy at each core due to message exchanges among cores.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we presented the design and analysis of core-assisted routing scheme for Opportunistic Networks (ONs). We introduced core nodes as specially designated nodes within the ON that facilitate the exchange of messages in the system. Besides, such core nodes can also be assigned to provide critical network management functions, such as security or group membership. Using analytical models, we provided performance analysis of core-assisted routing schemes for basic performance metrics such as message delay, message delivery ratio, and buffer occupancy when there is a time a time constraint for message delivery. Simulation results show that our model and analysis are accurate.

In the future, we would like to consider different methods to reduce buffer occupancy and message transmission by having node share information about delivered messages. We also

plan to extend the core-assisted routing approach to multicast routing in ONs.

REFERENCES

- [1] Scott Burleigh and Kevin Fall. Delay tolerant networking: An approach for interplanetary internet. *IEEE Communications Magazine*, June 2003.
- [2] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, and L. Wei. The pim architecture for wide-area multicast routing. *IEEE/ACM Trans. on Networking*, 4:153–162, 1996.
- [3] L. Chen et. al. A hybrid routing approach for opportunistic networks. In *SIGCOMM workshop on Challenged networks*, pages 213–220, Sept. 2006.
- [4] Leguay et. al. Opportunistic content distribution in an urban setting. In *SIGCOMM workshop on Challenged networks*, pages 205–212, Sept. 2006.
- [5] Kevin Fall. A delay-tolerant network architecture for challenged internets. Karlsruhe, Germany, August 2003. ACM SIGCOMM’03.
- [6] J. Garcia-Luna-Aceves and E. Madruga. The Core-Assisted Mesh Protocol. *JSAC*, 12(8):1380–1394, 1999.
- [7] R. Groenevelt, P. Nain, and G. Koole. The message delay in mobile ad hoc networks. In *Performance*, 2005.
- [8] Matthias Grossglauser and David N. C. Tse. Mobility increases the capacity of ad-hoc wireless networks. In *INFOCOM*, pages 1360–1369, 2001.
- [9] K. A. Harras, K. C. Almeroth, and E. M. Belding-Royer. Delay tolerant mobile networks: Controlled flooding in sparse mobile networks. In *Proceedings of Networking*, Waterloo, Ontario, Canada, May 2005.
- [10] Sushant Jain, Kevin Fall, and Rabin Patra. Routing in a delay tolerant network. ACM SIGCOMM’04, August 2004.
- [11] R. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: Modeling a three-tier architecture for sparse sensor networks. IEEE SNPA, 2003.
- [12] T. Small and Z. J. Haas. Resource and performance tradeoffs in delay-tolerant wireless networks. In *WDTN ’05*, pages 260–267, 2005.
- [13] T. Spyropoulos, K. Psounis, and C. Raghavendra. Single-copy routing in intermittently connected mobile networks. In *Sensor and Ad Hoc Communications and Networks*, pages 235–244. IEEE, Oct 2004.
- [14] T. Spyropoulos, K. Psounis, and C. Raghavendra. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. Philadelphia, PA, USA, August 2005. ACM SIGCOMM’05 Workshops.
- [15] T. Spyropoulos, K. Psounis, and C. Raghavendra. Performance analysis of mobility-assisted routing. pages 49–60. ACM Mobihoc 2006, 2006.
- [16] M. M. Bin Tariq, M. Ammar, and E. Zegura. Message ferry route design for sparse ad hoc networks with mobile nodes. In *MobiHoc*, 2006.
- [17] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Duke University, 2000.
- [18] J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. *INFOCOM*, 2003.
- [19] X. Zhang, G. Neglia, J. Jurose, and D. Towsley. Performance modeling of epidemic routing. Technical Report CMPSCI 2005-44, UMass, 2005.
- [20] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. *MobiHoc*, May 2004.