RelayCast: Scalable Multicast Routing in Delay Tolerant Networks

Uichin Lee†, Soon Young Oh†, Kang-Won Lee§, Mario Gerla†
†UCLA §IBM Research
†{uclee,soonoh,gerla}@cs.ucla.edu, §kangwon@us.ibm.com

***Abstract*— Mobile wireless networks with intermittent connectivity, often called Delay/Disruption Tolerant Networks (DTNs), have recently received a lot of attention because of their applicability in various applications, including multicasting. To overcome intermittent connectivity, DTN routing protocols utilize mobility-assist routing by letting the nodes carry and forward the data. In this paper, we study the scalability of DTN multicast routing. As Gupta and Kumar showed that unicast routing is not scalable, recent reports on multicast routing also showed that the use of a multicast tree results in a poor scaling behavior. However, Grossglauser and Tse showed that in delay tolerant applications, the unicast routing overhead can be relaxed using the two-hop relay routing where a source forwards packets to relay nodes and the relay nodes in turn deliver packets to the destination via "mobility," thus achieving a perfect scaling behavior of $\Theta(1)$. Inspired by this result, we seek to improve the throughput bound of wireless multicast in a delay tolerant setting using mobility-assist routing. To this end, we propose RelayCast, a routing scheme that extends the two-hop relay algorithm in the multicast scenario. Given that there are n_s sources each of which is associated with n_d random destinations, our results show that RelayCast can achieve the throughput upper bound of $\Theta(\min(1, \frac{n}{n_s n_d}))$. We also analyze the impact of various network parameters and routing strategies (such as buffer size, multi-user diversity among multicast receivers, and delay constraints) on the throughput and delay scaling properties of RelayCast. Finally, we validate our analytical results with a simulation study.**

I. INTRODUCTION

Protocols that can withstand intermittent connectivity caused by mobility and low node density, often called Delay Tolerant Network (DTN) Protocols, are becoming increasingly important in disruptive Mobile Ad Hoc Network (MANET) scenarios such as inter-vehicle communications, "pocket switched" personal networking among pedestrians, tactical communications in the battlefield and disaster recovery operations. In those scenarios, there has been a growing interest in DTN multicast protocols that enable distribution of situational data to multiple receivers, such as real-time traffic information reporting, diffusion of participatory sensor data, or software patch over multiple devices, in spite of the disruptive nature and intermittent connectivity of tactical MANETs.

Routing in a DTN is challenging because conventional MANET protocols can withstand only very short term path interruptions; they systematically fail when the network stays disconnected for a prolonged time. In favorable motion conditions, DTN routing protocols can overcome such intermittent connectivity by exploiting a *mobility-assist routing* strategy: nodes receive, hold in storage, and wait for opportunities to transfer packets to remote nodes. If the characteristics of a network (e.g., node mobility and traffic pattern) are known in advance, we can design "predictive" unicast/multicast routing algorithms that efficiently route packets over a *time-varying connectivity* graph [20], [46]. In practice, however, only limited information is available about network connectivity as a function of time. In view of this, researchers have investigated meaningful mobility statistics that allow one to make a better routing decision such as encounter history [29], [37], mobility pattern space [26] and social networking [6]. In addition, redundancy and coding techniques have been used to further improve reliability and reduce latency of DTN routing [38], [36], [19], [40].

Scalability is a very important metric when designing a routing protocol both in MANETs and in DTNs. For unicast, the scaling behavior is well understood. In their seminal work, Gupta and Kumar [16] showed that the scalability of *wireless multi-hop routing* is limited; in fact, in a wireless network with n static nodes, each engaged in a data transfer to a random destination the per node throughput decays as $\Theta(1/\sqrt{n \log n})$.[1] Realizing that the increasing hop length of a path is the key limiting factor when the number of nodes increases, Grossglauser and Tse [15] showed that under random mobility assumptions a two-hop relay routing strategy, a mobility-assisted routing protocol that exploits mobility and carry-forward to reduce number of hops can achieve $\Theta(1)$ throughput per node, thus exhibiting a perfect scaling behavior. However, the throughput improvement comes at the cost of increased delay. This result has been followed by a flurry of research activities that tried to characterize the delay/capacity relationship as a function of node mobility [11], [30], [34], [12], [44]. Due to the increased end-to-end delay, the need to buffer the packets until delivery to destination has prompted the study of the impact of finite node buffers on performance [17].

The scaling throughput properties in static wireless networks were recently generalized also to multicast and broadcast [33], [27], [41]. Assuming that there are n_s sources each

This research is supported through participation in the International Technology Alliance sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defense under Agreement Number W911NF-06-3-0001, and; by ARMY MURI under funding W911NF0510246.

[1]Recall that (i) $f(n) = O(g(n))$ means that $\exists c$ and $\exists N$ such that $f(n) \leq cg(n)$ for $n > N$ (i.e., asymptotic upper bound); (ii) $f(n) = \Omega(g(n))$ means that $\exists c$ and $\exists N$ such that $f(n) \geq cg(n)$ for $n > N$ (i.e., asymptotic lower bound); (iii) $f(n) = \Theta(g(n))$ means that $f(n) \in O(g(n)) \cap \Omega(g(n))$ (i.e., asymptotic tight bound); (iv) $f(n) = o(g(n))$ means that $\lim_{n\to\infty} f(n)/g(n) = 0$ (i.e., asymptotic insignificance); and (v) $f(n) = \omega(g(n))$ means $\lim_{n\to\infty} f(n)/g(n) = \infty$ (i.e., asymptotic dominance).

of which is associated with n_d random destinations and that the packets are delivered on multicast trees, the throughput per multicast source is $\Theta(\frac{\sqrt{n}}{n_s\sqrt{\log n}}\frac{1}{\sqrt{n_d}})$. The penalty of using a multicast tree is high; namely, it corresponds to a factor of $\sqrt{n_d}$ throughput decrement. When the number of multicast receivers is above a threshold value of $\Omega(n/\log n)$, multicasting scales as network wide broadcasting. Thus, its throughput becomes $\Theta(1/n_s)$ [39], [22]. This follows from the fact that above the threshold the multicast protocol can fully benefit from the wireless broadcasting effects [22], [41].

In this paper, we seek to improve the pathological throughput bound of wireless multicast using a mobility-assist routing algorithm. Namely, we propose RelayCast, a routing scheme that extends the Grossglauser and Tse's two-hop relay strategy by requiring that a relay node be responsible for delivering packets directly to each multicast receiver. This extended protocol is analyzed under the assumption that inter-contact time of an arbitrary pair of nodes follows an exponential distribution with rate λ. We compare throughput and delay properties of RelayCast with those of conventional multicast. In favorable mobility conditions, RelayCast offers two main benefits: it improves throughput scalability with increasing number of nodes, and; it provides reliable delivery even in DTN scenarios with intermittent connectivity. We then analyze the impact on RelayCast throughput performance of various network and routing parameters including buffer size, multicast receiver relay, and delay constraints.

The following is the preview of the key contributions of this paper.

- We find the throughput upper bound of DTN multicast routing and propose RelayCast, a two-hop relay based DTN multicast routing protocol, that achieves the upper bound, namely throughput = $\Theta(n\lambda)$ for the case $n_sn_d = O(n)$ and $\Theta(\frac{n^2\lambda}{n_sn_d})$ for the case $n_sn_d = \omega(n)$, or simply $\Theta(\min(n\lambda, \frac{n^2\lambda}{n_sn_d}))$. We also show that the delay of RelayCast is $\Theta(\frac{\log n_d}{\lambda})$. In particular, for a given $\lambda = \Theta(1/n)$, the throughput and delay of RelayCast are $\Theta(\min(1, \frac{n}{n_sn_d}))$ and $\Theta(n \log n_d)$ respectively. We provide a throughput/delay comparison between RelayCast and conventional multicast [33], [27].
- We show that RelayCast requires buffer space per multicast source of size $\Theta(nn_d)$ for the case $n_sn_d = O(n)$ and of size $\Theta(\frac{n^2}{n_s})$ for the case $n_sn_d = \omega(n)$; the aggregate buffer space to sustain all flows is bounded by $\Theta(n^2)$. Given finite buffer space of size K, the throughput per multicast source of RelayCast is reduced to $\Theta(\frac{K\lambda}{n_d})$.
- We prove that multicast receiver relay where multicast receivers cooperate in delivering the packets cannot improve the delay with respect to RelayCast, unless the number of multicast receivers scales as $n_d = \Theta(n)$. In this case, we show that there is an optimal multiple message "gossiping" protocol that can reduce the delay to $\Theta(\frac{\log n}{n\lambda})$. We also identify a scalability problem due to packet reconciliation overhead in some of the schemes proposed in the literature [8], [10].
- We compute throughput-delay trade-offs where throughput is traded for delay. In particular, we analyze RelayCast with k-copy replication and show that its throughput and delay are $O(\min(\frac{n\lambda}{k}, \frac{n^2\lambda}{kn_sn_d}))$ and $O(\frac{k\log k+n\log n_d}{nk\lambda})$ respectively.

The rest of the paper is organized as follows. In Section II, we present the network model. In Section III, we compute throughput and delay of RelayCast and compare with conventional multicast. In Section IV, we formally investigate various DTN multicast routing design parameters and their impacts on the scaling properties. In Section V, we validate our results via simulations. Finally, we present the conclusion in Section VII.

II. Network Model

In this section, we review communication model and traffic patterns; we define throughput and delay, and; we introduce a simple mobility model all of which will lead represent a DTN in general.

Communication Model and Traffic Patterns: We use the protocol model to abstract interference between transmissions [16]. Suppose that node i transmits to node j. Node j receives the transmission successfully if every other node that transmits simultaneously is at a distance of at least $(1+\Delta)r(n)$ from j where Δ is some positive number and $r(n)$ is the radio range. In the network, n_s nodes are randomly selected as multicast sources and each of these sources is associated with n_d multicast receivers, thus making a total of n_sn_d source-destination pairs in the system. We assume that a membership is fixed and for a given source, every node maintains a list of members.

Definition of Throughput and Delay: For a given scheduling algorithm π, a throughput $\gamma > 0$ is said to be feasible/achievable if every node can send at a rate of γ bits per seconds to its chosen destination. Let $T^\pi(n)$ denote the maximum feasible per-node throughput under scheduling algorithm π. The delay of a packet in a network is the time for a packet to reach the destination after it leaves the source. Let $D^\pi(n)$ denote the average packet delay for a network with n nodes under scheduling algorithm π. Note that a scheduling algorithm is *stable* if the rate $T^\pi(n)$ is satisfied by all users such that one's queue does not grow infinity; i.e., $D^\pi(n)$ is bounded.

Modeling Mobility: DTN protocols leverage node mobility as a means of data delivery (i.e., *carry-and-forward*) and thus, the performance mainly depends on the encounter pattern. In this paper, we describe the mobility model using the pairwise inter-contact time, i.e., the time interval between two successive encounters of a pair of nodes. For analysis, we consider a class of random mobility models where each node independently makes decision on its movement, e.g., each node independently chooses a a random *direction* (Random Direction). Groenevelt et al. showed that the inter-contact stochastic process of these mobility models can be captured using an independent homogeneous Poisson process with some meeting rate λ [14], [13]. In other words, inter-contact time distributions of any pairs are exponentially distributed with rate λ. This concept can be generalized using heterogeneous meeting rates with λ_{ij} for $i, j = 1, \cdots, n$. We present the Theorem 4.2.1 from [13] to provide a basis for estimating the λ value for different mobility models.

Theorem 1: Given that two nodes move randomly in a 1×1 unit area ($1\times1m^2$) with the average speed v, if the transmission range $r \ll 1$ and the position of a node at time $t+\Delta$ is independent of its position at time t for small Δ, then the inter-contact time between two nodes is exponentially distributed with parameter $\lambda = \alpha rv$ where α is a mobility model dependent constant.

In various empirical studies, the inter-contact time distribution has been reported to follow an exponential distribution in real-life mobility patterns. Conan et al. showed that several mobility traces contain significant fraction of contact pairs following exponential distributions [5]. For instance, in the Dartmouth College WiFi trace, out of 13,482 pairs 62.3% pairs have been found to follow an exponential distribution. Karagiannis et al. found an invariant property that there is a time granule in the order of half a day, up to which the distribution of inter-contact time is well approximated by a power law and beyond it decays exponentially [21]. They also found that the aggregate inter-contact distribution does not deviate significantly from the individual pairwise inter-contact time distribution. In general, when a mobility model is defined in a finite domain, it has been mathematically proven that the inter-contact time distribution has an exponential tail [3].

DTN Model: We model an arbitrary DTN in a unit area of (1×1) using the pairwise inter-contact rate $\lambda = \Theta(rv)$ where r is radio range and v is speed. We note that it is possible to map any delay tolerant network to a wireless network in a unit area by appropriately scaling the radio range and average speed. In our study, we consider two cases: (a) when λ is given and fixed and (b) when λ scales according to r and v.

When λ is given, Theorem 1 shows that the contact rate is independent of the number of nodes. As shown later, this allows us to predict the performance of DTN as a function of the number of nodes in the network. However, increasing the number of nodes over a certain limit will reduce the effective capacity due to wireless interference. Also, the node increase will eventually change the connectivity of the network from a DTN state to a fully connected state.[2] Thus, in order for the network to remain in a delay tolerant state and maximize the throughput, the number of nodes should be bounded. We can identify this bound as follows. Assume that the nodes are uniformly distributed on a unit square. The radio range determines the number of simultaneous transmissions, and thus the network-wide aggregate throughput. Since the number of transmissions is approximately the same as the total number of non-overlapping circles with radius r that fills 1×1 area, the network-wide aggregate throughput $\mathcal{T}$ is bounded by $\Theta(1/r^2)$. Therefore, the aggregate throughput can be expressed in terms of λ: i.e., $\mathcal{T} \leq \Theta(1/r^2) = \Theta(1/\lambda)$. For a DTN with the radio range r, the upper bound of the per-node throughput can be maximized, when the number of nodes is in the same order as the aggregate throughput, i.e., $\Theta(1/r^2) = \Theta(n)$ and thus, $r = \Theta(1/\sqrt{n})$. In this paper, we analyze more general scaling behavior with the radio range of $O(1/\sqrt{n})$.

On the other hand, if λ scales with the node speed and the radio range (which are functions of the number of nodes), we have $\lambda = \Theta(rv)$. In this case, we scale the node speed based on the radio range such that the contact duration of two nodes is constant as in [7], [11], [34]. Unless otherwise mentioned, we assume that the radio range is $r = O(1/\sqrt{n})$, and the speed $v = O(1/\sqrt{n})$ (thus, $\lambda = O(1/n)$). We then can easily show that the node density within one's radio range is bounded by $\Theta(1)$. Note that Grossglauser and Tse showed that when we scale the radio range as $r = \Theta(1/\sqrt{n})$, a class of DTNs with $\lambda = \Theta(1/n)$, we can achieve the throughput of $\Theta(1)$ using the two-hop relay "unicast" routing protocol. We assume that the network area is partitioned into C non-overlapping cells with size $s_n \times s_n$ where we have $s_n = 1/\sqrt{n}$ to have the node density per cell $O(1)$.

In this paper, we slightly abuse the asymptotic notation for simplicity. For instance, when we denote that the throughput per multicast source of RelayCast is $\Theta(\min(n\lambda, \frac{n^2\lambda}{n_s n_d}))$, this statement is always true only when λ scales with n. However, when λ is fixed, it is true only when $n \leq 1/\lambda$. This conditional rule applies to all asymptotic notations in this paper.

III. Throughput and Delay of DTN Multicast Routing

We derive the upper bound on the throughput of DTN multicast routing. We then proceed to present RelayCast, a 2-hop relay-based DTN multicast routing protocol. We analyze the throughput and delay of RelayCast and show that RelayCast achieves the throughput upper bound. Finally, we compare the throughput and delay of RelayCast with those of conventional wireless multi-hop multicast.

A. Multicast Throughput Upper Bound in DTNs

The below theorem shows the throughput upper bound of DTN multicast routing where we have n_s multicast sources and n_d multicast receivers.

Theorem 2: The throughput upper bound of DTN multicast is $\Theta(\min(n\lambda, \frac{n^2\lambda}{n_s n_d}))$.

Proof: We use a derivation that is similar to that in [16]. In the network, n_s nodes are randomly selected as multicast sources and each of these sources is associated with n_d multicast receivers. Consider a bit b originating at a source. In our network setting, there are a constant number of nodes in each cell.[3] The chance of transmission is equally shared by c interfering nodes under the protocol model [24]. Thus, the minimum number of transmissions required to deliver a bit b to n_d destinations is $\Theta(n_d)$, even with broadcasting effects. Under any scheduling algorithm, we need $H(b) = \Omega(n_d)$ transmissions to deliver a bit b. For a given time slot, node i encounters a random node with probability $n\lambda$. Considering the interference, the node can transmit with probability $n\lambda/c$. This transmission opportunity is denoted as an indicator random variable S_i. The total number of simultaneous transmissions is given as $S = \sum_{j=1}^{n} S_j$. Its expectation is $\mathbb{E}[S] = n\mathbb{E}[S_i] = n^2\lambda/c$. Each source generates bits with rate $T(n)$. For a given period τ, the total number of bits generated in the network is $n_s T(n)\tau$. The total number of hops required to support

[2]A network is connected with high probability if its transmission range is set to $\Theta(\sqrt{\log n/n})$ [16]. Thus, for a given transmission range, we can find the number of nodes that make the network connected.

[3]For a given contact, the number of interfering nodes is given as $\Theta(n/r^2) = \Theta(1)$

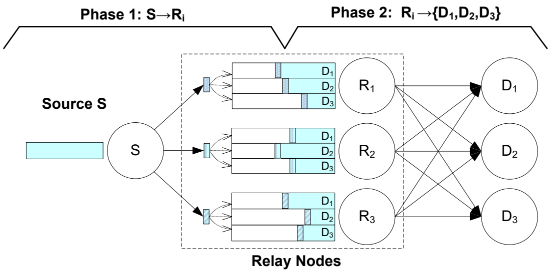

Fig. 1. RelayCast: DTN multicast based on 2-hop relay. Relay node R_i delivers a packet to all the multicast receivers. Note that receiver D_i can also be a relay node.

these bits during time interval τ is $n_sT(n)\tau H(b)$. This is bounded by the total number of feasible transmissions in the network during time interval τ that is τS. Hence, we have $n_sT(n)\tau H(b) \leq \tau S$. By substituting $H(b)$ and S, we have $T(n) \leq \frac{n^2\lambda}{cn_sn_d}$ and thus, $T(n) = O(\frac{n^2\lambda}{n_sn_d})$. The DTN multicast throughput is bounded by its unicast throughput, especially when $n_sn_d \leq n$. Since the unicast throughput is a special case of multicast (i.e., $n_s = n$ and $n_d = 1$), the throughput is given as $O(n\lambda)$. Thus, we have $T(n) = O(\min(n\lambda, \frac{n^2\lambda}{n_sn_d}))$. ■

To contrast the multicast with the unicast, let us take the number of source destination pairs to be the same in both case; i.e., $n_sn_d = n$. We take $n_d = n^{1-\epsilon}$ and $n_s = n^\epsilon$ where $0 \leq \epsilon \leq 1$. As long as $n_sn_d = n$ is satisfied, the throughput is essentially the same as unicast throughput, i.e., $\Theta(n\lambda)$. As $\epsilon \to 1$ the multicast is the same as unicast whereas $\epsilon \to 0$, it becomes a delay tolerant broadcasting.

B. RelayCast: 2-Hop Relay-based DTN Multicast Routing

We present a DTN multicast protocol called RelayCast whose operations are based on 2-hop relay DTN routing. For each time slot a cell becomes active if it contains at least a pair of nodes that are within the radio range of each other. In each active cell, we randomly select a pair of nodes and perform either of the following operations. In Phase 1 (Relay), the multicast source sends a new packet to a relay node. The relay node could be one of the multicast receivers. In Phase 2 (Delivery), if there is a multicast receiver that has not received a packet yet, a relay node delivers the packet. The overall procedure is illustrated in Figure 1. Note that a relay node has a separate queue for each multicast destination and replicates an incoming packet to each of the relay queues (i.e., n_d replicas).

Theorem 3: The throughput of RelayCast per multicast source is $\Theta(\min(n\lambda, \frac{n^2\lambda}{n_sn_d}))$.

Proof: Consider a multicast stream: source s and a set of destinations d_i for $i = 1, \cdots, n_d$. The throughput per source is $\Theta(n\lambda)$ if each destination d_i can achieve $\Theta(n\lambda)$, which we will show in the following. During a small time interval Δt, a random node j encounters the destination d_i with probability $\lambda\Delta t + o(\Delta t)$. In our network setting, there are a constant number (c) of nodes in each cell under the protocol model. Since the chance of transmission is equally shared by c interfering nodes, node j can successfully deliver a packet with the probability $\lambda\Delta t/c$. Recall that we have n_s sources each of which is associated with n_d destinations chosen randomly. The probability that a node chooses a random node as a destination is $p = n_d/n$. We want to know how many sources out of $n_s - 1$ will choose node d_i as a destination as well. The probability that ℓ sources choose a certain node as a destination is given as $\binom{n_s-1}{\ell}p^k(1-p)^{n_s-1-\ell}$, and on average there will be $\frac{(n_s-1)n_d}{n}$ sources. Let n_x denote the total number of sources competing for the limited resources including the source s. Then, we have $n_x = \frac{(n_s-1)n_d}{n} + 1$. When $n_sn_d = O(n)$, we have $n_x = \Theta(1)$; and when $n_sn_d = \omega(n)$, we have $n_x = \omega(1)$. Assuming that each source equally shares the overall transmission opportunities, this packet belongs to a source i with probability $1/n_x$. Here, we are interested in the event that the receiver d_i is scheduled to receive node i's packet at time t. Let an indicator random variable $M_i(\Delta t, n)$ denote this event. Since d_i can meet any of the relay nodes, we have:

$$Pr\{M_i(\Delta t, n) = 1\} \tag{1}$$

$$= \sum_{j=1, j\neq d_i}^{n} Pr\{\text{node } j \text{ delivers a packet during } \Delta t\} \tag{2}$$

$$\approx \frac{(n-1)\lambda\Delta t}{n_x c} \tag{3}$$

Thus, the throughput is given as:

$$T_{d_i}(n) = \frac{\mathbb{E}[M_i(t,n)]}{\Delta t} = \frac{(n-1)\lambda\Delta t}{n_x c}\frac{1}{\Delta t} \tag{4}$$

$$= \begin{cases} \Theta(n\lambda), & n_sn_d = O(n) \\ \Theta(\frac{n^2\lambda}{n_sn_d}), & n_sn_d = \omega(n) \end{cases} \tag{5}$$

The above cases can be simplified as $\Theta(\min(n\lambda, \frac{n^2\lambda}{n_sn_d}))$. ■

When the radio range is scaled appropriately such that $r = \Theta(1/\sqrt{n})$, (and therefore $\lambda = \Theta(1/n)$), the throughput per source is given as $\Theta(\min(1, \frac{n}{n_sn_d}))$. If the number of source-destination pairs is less than $n_sn_d = O(n)$, the throughput per multicast source is $\Theta(1)$ as in two-hop relay where there are n source destination communication pairs.

Theorem 4: The average delay of RelayCast is $\Theta(\frac{\log n_d}{\lambda})$.

Proof: We find the average delay to deliver a packet to all n_d receivers. The relay node encounters the first receiver with rate $n_d\lambda$ and the average delay is $\frac{1}{n_d\lambda}$. At that moment, there are still $n_d - 1$ receivers waiting for the packet. By the memoryless property, we can simply treat them as if they just begin. Thus, the average time to meet the second receiver is simply $\frac{1}{(n_d-1)\lambda}$. By repeating this process, we have:

$$E[D] = \frac{1}{n_d\lambda} + \frac{1}{(n_d-1)\lambda} + \cdots + \frac{1}{\lambda} \tag{6}$$

$$= \frac{1}{\lambda}\sum_{i=1}^{n_d}\frac{1}{i} \tag{7}$$

$$= \frac{1}{\lambda}\left(\log n_d + \gamma + O\left(\frac{1}{n_d}\right)\right) \tag{8}$$

where γ is Euler's constant. Thus, $D(n) = \Theta(\frac{\log n_d}{\lambda})$. The packet buffering at each node will incur additional delay in the end-to-end delay computation. However, the queueing delay

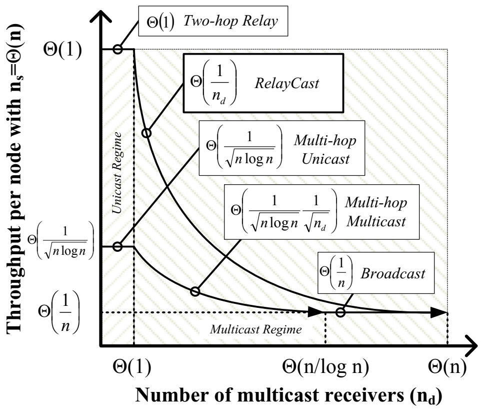

Fig. 2. Throughput scaling result comparison

increases the average delay of each step with a constant factor.[4] Thus, the order of the average delay does not change. ■

C. Comparison with Multi-hop Wireless Multicast Routing

We compare the throughput/delay scaling of conventional multi-hop wireless multicast routing with that of RelayCast. For this we first review the throughput scaling of multi-hop wireless multicast routing where the radio range scales with $\Theta(\sqrt{\log n/n})$. The following theorem from [27] shows the throughput of multi-hop wireless multicast routing. Similar results have been reported in [33], [41].

Theorem 5: The throughput per multicast source is upper bounded by $\Theta\left(\frac{\sqrt{n}}{n_s\sqrt{\log n}}\frac{1}{\sqrt{n_d}}\right)$ when $n_d = O(\frac{n}{\log n})$ and by $\Theta(\frac{1}{n_s})$ when $n_d = \Omega(\frac{n}{\log n})$.

As shown in Theorem 2, the key factor of determining the throughput upper bound is the number of transmissions (or hop count) $H(b)$. Du et al. [9] showed that the Euclidean distance of a minimum spanning tree covering n_d nodes is $\Theta(\sqrt{n_d})$, and thus, we have $H(b) = \Theta(\sqrt{n_d}/r(n))$. Interestingly, if the number of receivers is greater than $\Omega(\frac{n}{\log n})$, multicast becomes a network-wide broadcast whose throughput per node is $\Theta(1/n_s)$ [22], [39].[5]

In Figure 2, we summarize the throughput per node with $n_s = \Theta(n)$ as a function of the number of multicast receivers. Unlike conventional multi-hop wireless multicast routing, the throughput per node of RelayCast is $\Theta(\min(n\lambda, \frac{n^2\lambda}{n_s n_d}))$; in particular, when the radio range scales as $\Theta(\min(1, \frac{n}{n_s n_d}))$, i.e., $\lambda = \Theta(1/n)$. Since the number of sources is $n_s = \Theta(n)$, the throughput per node of RelayCast is $\Theta(1/n_d)$. The throughput per node of conventional multi-hop multicast is $\Theta\left(\frac{1}{\sqrt{n\log n}}\frac{1}{\sqrt{n_d}}\right)$ when $n_d = o(\frac{n}{\log n})$. If the number of receivers is $n_d = \Omega(\frac{n}{\log n})$, the throughput per node is $\Theta(1/n)$. The throughput per source of RelayCast is better than that of conventional multi-hop wireless multicast routing. When the number of receivers is $n_d = \Theta(n)$, the throughput per node of wireless broadcast is the same as that of RelayCast.

We now find the delay of conventional multi-hop multicast routing. El Gamal et al. [11] showed that the average delay of unicast routing is proportional to the number of hops to deliver a packet: the average distance between a random pair of nodes is $\Theta(1)$, and the average number of hops is simply $\Theta(1/r(n)) = \Theta(\sqrt{n/\log n})$. Given this, we realize that the average delay of multicast routing is no different than that of unicast, because a packet can be delivered in parallel along different paths of the multicast tree, and the depth of a multicast tree is asymptotically the same as that of unicast routing.[6] While the delay of conventional multi-hop multicast routing is independent of the number of receivers, that of RelayCast is a function of the number of receivers, increasing logarithmically as $\Theta(\log n_d/\lambda)$. Given a DTN of $\lambda = 1/n$, the delay of RelayCast is $\Theta(n \log n_d)$ that is much greater than that of conventional multi-hop multicast routing. In Section IV-C, we investigate the delay-throughput trade-offs of RelayCast.

IV. Protocol Design Parameter Analysis

A. Buffer Requirements and Impact of Finite Buffer

In RelayCast, a relay node maintains a queue for each multicast receiver and an incoming packet is replicated n_d times. As an alternative, we can emulate the multi-queue scheme using a single queue with per packet delivery status bookkeeping (or a simply list of nodes that have received a certain packet). A relay node scans the queue from the head, and selects a packet that has not yet been delivered to the encountered destination node. When a packet is delivered to all the multicast receivers, it is removed from the queue.

We find the average buffer space for both cases to sustain the maximum throughput of RelayCast, i.e., $\Theta(\min(n\lambda, \frac{n^2\lambda}{n_s n_d}))$ using Little's law: the product of per node throughput and average packet lifetime. Since the packet life time is equal to the average delay, the average number of packets for a given multicast flow (or the average number of "in-flight" packets) is determined by the throughput-delay product in a DTN. The buffer requirement of RelayCast can be computed as follows. We first analyze the multiple queue scheme. When we have $n_s n_d = O(n)$, the incoming rate to a relay node is $\Theta(\lambda)$. Since the average delay of a packet is $\Theta(1/\lambda)$, the average number of packets in a queue is $\Theta(1)$. There are $\Theta(n)$ relay nodes each of which has n_d queues. Thus, the aggregate buffer space required to sustain a multicast flow is $\Theta(nn_d)$. Similarly, when we have $n_s n_d = \omega(n)$, the aggregate buffer space is $\Theta(\frac{n\lambda}{n_s n_d}) \times \frac{1}{\lambda} \times nn_d = \Theta(\frac{n^2}{n_s})$.

Now we consider the single queue scheme with per packet delivery status bookkeeping. When we have $n_s n_d = O(n)$,

[4]Note that the queueing behavior can be modeled using the standard M/M/1 queue. In a relay node, a packet can be located in a random position of each queue. In the beginning, since it has not delivered to any of n_d destinations, we consider all n_d queues. The aggregate service rate is $n_d\lambda$. The key is that the utilization of the queueing system must be $\rho < 1$ in order to be stable. The average sojourn time is simply given as $\mathbb{E}[W] = \frac{1}{(1-\rho)n_d\lambda}$ [23]. Thus, the queueing delay increases the delay only a constant factor. After the packet is delivered to one of the receivers, we then consider $n_d - 1$ queues in the same manner.

[5]In this case, multicast receivers will cover at least a constant fraction of the 1×1 network. To be precise, because there are $1/r(n)^2$ cells with at least one receiver [27], the total area size covered by these nodes is constant, i.e., $\Theta(1/r(n)^2 \times r(n)^2) = \Theta(1)$. Thus, when the number of receivers is large enough (at least $\Omega(1/r(n)^2)$), the throughput put node is simply bounded by the broadcast capacity.

[6]Under the protocol model, the number of cells that interfere with any given cell is constant as shown in [24]. Thus, the interference among different paths does not affect the average delay.

the average number of packets in a queue is computed as $\Theta(\lambda) \times \Theta(\frac{\log n_d}{\lambda}) = \Theta(\log n_d)$, requiring much less space than the multiple queue scheme. However, since each packet requires $\Theta(n_d)$ space for bookkeeping and there are $\Theta(n)$ relay nodes, the aggregate buffer space to sustain a multicast flow is $\Theta(nn_d \log n_d)$.[7] In the same way, we can find the aggregate buffer space of the case $n_s n_d = \omega(n)$ as $\Theta(\frac{n\lambda}{n_s n_d}) \times \Theta(\frac{nn_d \log n_d}{\lambda}) = \Theta(\frac{n^2 \log n_d}{n_s})$. Thus, the single buffer scheme requires a factor of $\Theta(\log n)$ more buffer space than the multiple queue scheme. If we multiply n_s with the aggregate buffer space per source, we find the aggregate buffer space over all sources: $O(n^2 \log n_d)$ and $O(n^2)$ for single queue and multiple queue schemes respectively.

One may argue that in the single queue scheme, a node could perform packet reconciliation whenever it encounters another node, thus obviating the need of delivery status bookkeeping. To realize this, we must consider the overhead of packet reconciliation that is proportional to the average queue length per node. As there are n_s sources, the overhead is $\Theta(n_s \log n_d)$ for $n_s n_d = O(n)$ and $\Theta(\frac{n \log n_d}{n_d})$ for $n_s n_d = \omega(n)$. If the number of multicast receivers scales as the network size, so does the overhead of packet reconciliation. However, in our model we assume the constant contact duration, i.e., $\Theta(r(n)/v(n)) = \Theta(1)$, and thus, the single queue scheme is not feasible due to the overhead.

Impact of finite buffer: Given finite buffer space of size $n_s K$ in the network, we now want to find the throughput bound of RelayCast. We assume that a buffer replacement algorithm is *not* used such that a multicast source drops a packet (or fails to send a packet) if there is not enough space in a relay node. Note that it must have buffer space of size at least $\Theta(n_d)$ because of replication. Moreover, a freed buffer space will be taken by any of the n_s nodes with the same probability because of the uniform random encounter patterns. Therefore, we assume that buffer space is equally shared by n_s nodes, and a multicast flow is given buffer space of size K on average. Given this, the following theorem shows the throughput per multicast source of RelayCast with finite buffer.

Theorem 6: Given finite buffer space of size K in the network where $K = O(nn_d)$ for the case $n_s n_d = O(n)$ and $K = O(\frac{n^2}{n_s})$ for the case $n_s n_d = \omega(n)$, the throughput per multicast source of RelayCast is $\Theta(\frac{K\lambda}{n_d})$

Proof: Given finite buffer space of size K, it can support at most $\Theta(K/n_d)$ in-flight packets, because each packet needs to be replicated n_d times. Let us now investigate how this space of size $\Theta(K/n_d)$ can be distributed among relay nodes. When we have $K = O(nn_d)$, the throughput per node is $\Theta(n\lambda)$: a relay node can meet a destination node with rate $\Theta(\lambda)$ and there are n such nodes. The space of $K = O(K/n_d)$ should be equally shared by n potential relay nodes, and thus, a random relay node has a space for the destination with probability $\frac{K}{nn_d}$. When we have $\omega(nn_d)$, the throughput per node is $\Theta(\frac{n^2\lambda}{n_s n_d})$: a relay node encounters a destination node with rate $\Theta(\frac{n\lambda}{n_s n_d})$ – for a given destination node, the relay node uses its contract rate λ with probability $\frac{n}{n_s n_d}$. As there are $\Theta(n)$ potential relay nodes, the average number of relay nodes in the network that share the space of $O(K/n_d)$ is $\frac{n^2}{n_s n_d}$. Thus, a random relay node has a space for the destination with probability $\frac{K n_s}{n^2}$. Using the same proof technique as in Theorem 3, we find that the throughput per source is $\Theta(\frac{K\lambda}{n_d})$ in both cases. ■

B. Multicast Receiver Relay in DTN Multicast Routing

Since a packet is delivered to a set of multicast receivers, we want to consider cooperative relaying among multicast receivers or RelayCast with Multicast Receiver Relay (RelayCast-MRR). Unlike RelayCast where there is a single relay node for a given packet, RelayCast-MRR could have $O(n_d)$ relay nodes. It can potentially reduce the delay, yet preserve the throughput because the number of transmissions required to deliver a packet does not increase. However, we show that RelayCast-MRR cannot improve the delay unless the number of receivers scales as $\Theta(n)$ (i.e., network-wide broadcasting).

Theorem 7: If the number of multicast receivers scales as $n_d = o(n)$, RelayCast-MRR cannot improve the delay.

Proof: Packets arrives at a multicast receiver with the rate of $O(n\lambda)$ that is the throughput per source. To realize Multicast Receiver Relay, the receiver should be able to send received packets to other multicast receivers. As there are $\Theta(n_d)$ receivers, the multicast receiver relay rate is $\Theta(n_d\lambda)$. However, if we have $n_d = o(n)$, the arrival rate $O(n\lambda)$ is asymptotically greater than the service rate $\Theta(n_d\lambda)$. There will be infinite backlog of packets that need to be sent to other multicast receivers. Therefore, packets will never have a chance to be sent to other multicast receivers, and receivers cannot deliver received packets to other receivers. ■

From the above argument, we see that MRR is only feasible when we have $n_d = \Theta(n)$ (i.e., network-wide broadcasting) such that the incoming rate is the same order as the multicast receiver relay rate. We can achieve this using "multiple" message gossiping protocols where $\Theta(m)$ messages from a single source can be disseminated to n users in $\Theta(m + \log n)$ time steps in a distributed fashion [10], [32]. The key idea of optimal multiple message gossiping is to make every contact useful as follows. Due to the randomness of mobility patterns, it is expected that each node collects on average $N(t)$ messages out of M messages at time slot t. For a given encounter between u and v, we show that the probability that the meeting is useful is 1 with high probability. The meeting is useless if both u and v have the same set of messages. It happens with probability $p_{overlap} = 1/\binom{M}{N(t)}$ and we can easily show that $p_{overlap} < 1/M$ for $N(t) < M$. The probability that the meeting is useful is simply $1 - p_{overlap} > 1 - 1/M$. As M goes to infinity, we know that the probability converges to 1.

Theorem 8: The throughput upper bound is achievable with optimal multiple message gossiping.

Proof: To determine the throughput, we should find the effective meeting rate that is a fraction of the meeting rate useful for data transfer. We know that each relay node encounters a random node at the rate of $\Theta(n\lambda)$. Since there are $\Theta(n)$ potential relay nodes in the network, the aggregate rate is

[7]In practice, the bookkeeping overhead can be ignored in a network with a finite number of nodes.

$\Theta(n^2\lambda)$. Optimal multiple gossiping allows us to fully utilize meeting opportunities and the aggregate meeting rate can be fully used for data transfer. Thus, using the same argument as in Theorem 3, we can prove that the throughput upper bound is achievable. ■

For example, consider a single source scenario with $n_d = n$ multicast receivers. During n time contact time slots where the size of a contact time slot is $\Theta(\frac{1}{n\lambda})$, there will be $\Theta(n^2)$ transmission opportunities. During that period of time, a source has generated $\Theta(n\lambda \times 1/\lambda) = \Theta(n)$ packets on average. The total number of transmissions required to deliver these packets is given as $\Theta(n^2)$. Since we show that optimal multiple gossiping can fully utilize meeting opportunities, $\Theta(n^2)$ transmission opportunities can be used for data transfer. We can deliver n packets during n times slots and thus, the throughput can be computed as $\Theta(n\lambda)$.

Moreover, optimal multiple message gossiping can minimize the delay, yet fully utilize the contact opportunity by properly arranging data transfer opportunities. This allows us to decrease the delay by a factor of n: RelayCast takes $\Theta(\frac{\log n}{\lambda})$ whereas RelayCast-MRR using optimal multiple message gossiping takes $\Theta(\frac{\log n}{n\lambda})$ which is the average delay of broadcasting a single message [14]. Note that RelayCast-MRR actually improves the delay-throughput trade-offs that Neely et al. [30] reported: the throughput must be reduced by a factor of n in order to decrease the delay to $\Theta(\frac{\log n}{n\lambda})$.

The overall process can be best explained by a "pipeline" analogy, because multiple messages are being simultaneously transferred in the network. The number of in-flight packets can be viewed as the number of packets in the pipeline. Using the Little's results, we find that the number of in-flight packets is given as $\Theta(n \log n)$ and $\Theta(\log n)$ for RelayCast and RelayCast-MRR respectively. Thus, RelayCast has a pipeline of $\Theta(n \log n)$ stages, and RelayCast-MRR has a pipeline of $\Theta(\log n)$ stages. Each stage of a pipeline takes $\Theta(\frac{1}{n\lambda})$ on average. Given that there are infinite streams of packets from a multicast source, the throughput is mainly determined by the duration of stage. RelayCast has a factor of $\Theta(n)$ longer stages in its pipeline, but once the pipeline is fully loaded, a packet can be delivered in each time slot. Note that the optimal message gossiping disseminates a finite number of messages, m. For $\lambda = 1/n$, a single message broadcasting takes $\log n$ steps which is followed by $m-1$ steps to complete the rest of messages; thus, this will take $\Theta(m + \log n)$ steps [10], [32].

We now review the multiple message gossiping protocols, namely Randomized Multi-Message Gossip (RMMG) [10] and INTERLEAVE [32]. RMMG uses a coloring mechanism whereby each node has a unique color that indicates the message for which it has primary pushing responsibility, and an aging mechanism that limits the scope of primary message dissemination and enables pulling [10]. In RMMG, a node must send an M-bit vector in $\{0,1\}^M$, indicating the presence of the set of messages stored in the node, along with the list of colored packets. After packet reconciliation, a node either pushes a colored packet, or pulls a random packet that the node does not have.[8] As shown earlier, there are $M = \Theta(\log n)$ in-flight packets. Since the packet reconciliation overhead also scales as $\Theta(\log n)$, the contact duration must scale logarithmically to make it feasible.

Sanghavi et al. [32] proposed INTERLEAVE protocol that interleaves push and pull without any interference in a decentralized way as follows. Assume that every time a source transmits a packet, the sequence number will be incremented. For a given contact opportunity, INTERLEAVE operates as follows. The source pushes a new packet to the encountered node. Other nodes chooses the following two options at random: 1) a node pushes the highest number packet among the packets pushed by others, or 2) a node sends a pull request for the lowest numbered piece it does not have. INTERLEAVE shares the same idea as RMMG. The key difference is that INTERLEAVE has *removed* the packet reconciliation process based on the observation that after a new packet is out on the network for $\Theta(\frac{\log n}{n\lambda})$, it can be delivered to a constant fraction of the nodes, and thus, a pull request for a lower number packets is likely to succeed.

Note that Algebraic Gossiping (AG) also realizes optimal multiple message gossiping using random linear network coding [8]. Each message is represented an element in a vector space with the scalars in a finite field of appropriate size. For a given encounter, a node generates a coded message by combining all the coded messages collected thus far. If a node has all M "linearly independent" coded messages, it can recover the original messages. The key idea of AG is that network coding ensures that every coded message that one receives is useful (i.e., linearly independent from one's coded messages collected). In AG, the control overhead (i.e., a code vector in each packet) scales with the network size. Moreover, to decode the messages, we need to collect all $M = \Theta(\log n_d)$ messages and thus, the average delay is increased by a factor of $\Theta(\log n_d)$.

C. Throughput-delay Trade-offs

We show that the delay of RelayCast is $\Theta(\frac{\log n_d}{\lambda})$. Considering a DTN with $\lambda = 1/n$, the delay scales as $\Theta(n \log n_d)$. This is due to the fact that there is only a single relay node delivering a packet to all the multicast receivers. To reduce the delay, we consider the k-copy replication scheme where a packet is replicated to k relay nodes. We assume that each replica holder can replicate the packet as long as the total number of replicas is less than k (i.e., multi-hop replication) [38], [36]. For instance, Spyropoulos et al. proposed a binary spraying method [38]; i.e., a counter value that is initially set to k to generate k replicas is halved for each encounter and is distributed to nodes; a node finishes replication if its counter value reaches to zero. We can summarize the capacity-delay tradeoffs as follows.

Theorem 9: Given RelayCast with "k-copy" replication, the throughput per multicast source is $O(\min(\frac{n\lambda}{k}, \frac{n^2\lambda}{kn_s n_d}))$, and the average delay is $O(\frac{k \log k + n \log n_d}{nk\lambda})$.

Proof: The overall derivation of the throughput upper bound is very similar to Theorem 2, except that the hop

[8]Note that to minimize the average delay, we must select a packet with the smallest sequence number among the missing packets.

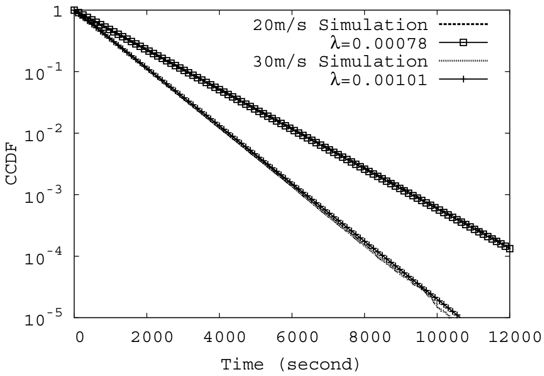

Fig. 3. Inter-contact time distribution

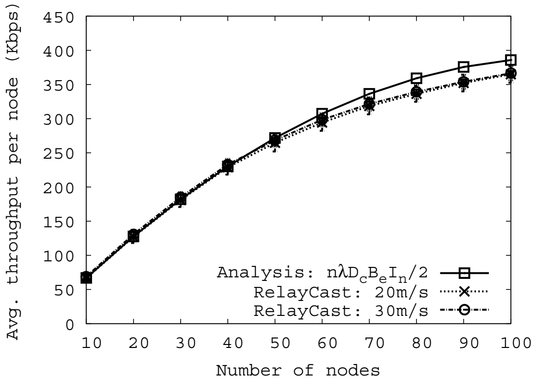

Fig. 4. Average throughput per source

length $H(b)$ is increased to k: given k-copy replication, each packet requires at least k transmissions. Thus, the throughput is reduced by a factor of k.

We now find the delay of k-copy replication. The delay upper bound can be represented as $E[D] \leq E[D_s] + E[D_d]$ where D_s represents the time to relay a packet to k distinct relay nodes (denoted as the spray step), and D_d represent the time to deliver a packet to all the multicast receivers using k distinct relay nodes (denoted as the delivery step). The average delay of the spray step can be calculated as follows. In the beginning, a source node relays a packet with rate $(n-1)\lambda$. After this, the rate is increased to $2(n-2)\lambda$ because there are two nodes spraying the packet. After k steps, the rate becomes $k(n-k)\lambda$. Thanks to the memoryless property, the average delay is simply represented as $E[D_s] = \sum_{j=1}^{k} \frac{1}{j(n-j)\lambda} = \frac{1}{n\lambda}\sum_{j=1}^{k}(\frac{1}{j} + \frac{1}{n-j}) = \Theta(\frac{\log k}{n\lambda})$. Moreover, the average delay of the delivery step can be easily calculated using Theorem 4. Given k relay nodes, the average delay is decreased by a factor of k, i.e., $\Theta(\frac{\log n_d}{k\lambda})$. Therefore, by replacing $E[D_s]$ and $E[D_d]$, we find the average delay of k-copy replication as $E[D] = O(\frac{k \log k + n \log n_d}{nk\lambda})$. ■

V. SIMULATIONS

We present the throughput and delay of RelayCast using QualNet v3.9.5, a packet level network simulator.

A. Simulation Setup

We use the random waypoint mobility model with 0 pause time and constant node speeds at 20m/s and 30m/s in a 5000m × 5000m region. We use 802.11b with 250m transmission range and 2Mbps transmission rate and use a two-ray ground path-loss propagation model. We use the Multicast Constant Bit Rate (MCBR) traffic in QualNet to measure the maximum throughput. We vary the number of nodes from 10 to 100, and warm up simulations for 10,000s. We implement RelayCast and compare its performance with analytical results. We also compare the results with On-demand Multicast Routing Protocol (ODMRP), a well-known multi-hop wireless multicast protocol [25]. For ODMRP, we set the refresh interval to be 3 seconds and the forwarder's lifetime to be 9 seconds. For inter-contact time measurement, the duration of a simulation is 100,000 seconds; for RelayCast performance measurement, we run simulation for 40,000 seconds. Reported results are the averages of 50 runs with different random seeds and are presented with the 95% confidence interval.

B. Results

We first present the pairwise inter-contact time of mobile nodes and show its Complementary Cumulative Distribution Functions (CCDF) with fitted exponential curves in Figure 3. The figure confirms that the inter-contact time of nodes with random waypoint movement closely matches exponential distribution. The mean inter-contact time is given as 1344.00s and 924.08s for the speed of 20m/s and 30m/s respectively.

We measure the per node throughput by increasing the CBR traffic rate (i.e., packets/sec). The size of a packet is 1500B. Heusse et al. showed that the channel utilization of 802.11b with a packet of size 1500B is 70% (denoted as u) [18]. For a given bandwidth B, the effective bandwidth is given as $B_e = Bu$. In Theorem 3, we showed that the throughput per source is $\Theta(n\lambda)$. In other words, a node encounters another node with rate $n\lambda$, which is a renewal process with the mean inter-renewal interval of $\frac{1}{n\lambda}$. For a given contact duration D_c, a node can transfer on average $D_c B_e$. We use a very simple interference model where interference reduces the throughput by a constant factor, and the frequency of interference is proportional to the number of nodes in the network. The average throughput is given as $n\lambda D_c B_e I_n$ where I_n denotes the degree of interference given n nodes. Now we want to know how throughput scales as the number of nodes increases when there is a single source that sends packets to $n-1$ multicast receivers in the network. If a random node is a pure relay node (not a multicast receiver), it can fully utilize its contact period delivering packets to the encounter receiver. However, in our case, every node other than the source is both a relay node as well as a multicast receiver and thus, the bandwidth is fairly shared by incoming and relaying traffic. Thus, the average throughput is given as $n\lambda D_c B_e I_n/2$. Figure 4 shows the measured throughput and analytical results. The figure shows that the analytic throughput model matches well with the simulated results, but they slightly deviate from each other as the number of nodes increases. We believe this gap can be reduced by using a more sophisticated interference model such as [31].

We compare the scalability of RelayCast with that of ODMRP. In particular, we evaluate the cases $n_s n_d \leq n$ where RelayCast can achieve the throughput of $\Theta(1)$. We increase the number of sources from 1 to 20 each of which has 5 random destinations. To find the best scenario of ODMRP, we use various MCBR rate ([20,200] pkts/s) with packet sizes of 512B

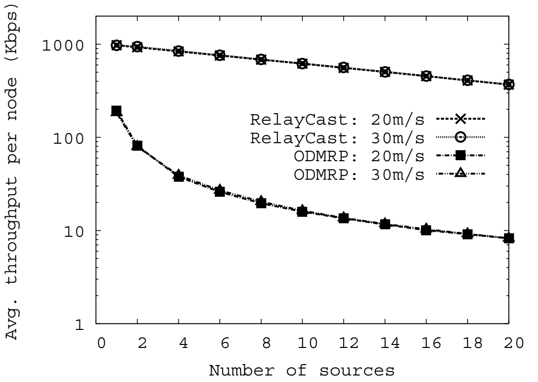

Fig. 5. Average throughput comparison between RelayCast and ODMRP

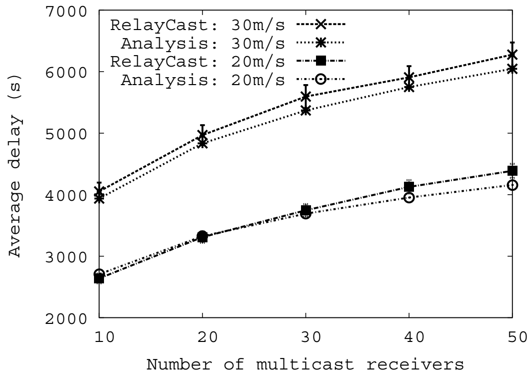

Fig. 6. Average delay of RelayCast

and 1024B, and different area sizes (750m $\times$ 750, 1000m $\times$ 1000m, and 1250m $\times$ 1250m). Our results show that the maximum throughput is attained when a 512B packet is sent at the rate of 200 pkts/s in an area of size 1000m $\times$ 1000m. Figure 5 reports the results. As the number of sources increases, the throughput of RelayCast slowly decreases, but the throughput of ODMRP decreases significantly. For instance, when the number of sources has increased from 1 to 2, the throughput of ODMRP is decreased from 183.6Kbps to 79.7Kbps, whereas that of RelayCast is reduced from 975.6Kbps to 926.8Kbps. This result confirms that RelayCast is a more scalable solution for multicast in DTN environments.

Finally, we investigate the average delay of RelayCast. We show how the average delay of RelayCast changes as the number of destinations increases. In order to measure the delay incurred by the protocol, we throttle down the sending rate at the source so that we can minimize the impact of queueing delay. This result is reported in Figure 6 along with analytic results from Theorem 4. The graph shows that our analytic results matches with simulation results fairly well. In general, the average delay increases, as the number of destinations increases. We also test the case with k-copy replication scheme, and our results confirm that replication can significantly reduce the delay.

VI. Related Work

DTN Multicast Routing Protocols: Zhao et al. proposed a set of DTN multicast semantic models regarding group membership and delivery interval, because group membership may change during message transfer due to large delay [46]. They incorporated various knowledge oracles such as contact and membership in the possible routing strategies (e.g., unicast, broadcast, tree/mesh). In practice, only limited information is available about network connectivity as a function of time. Ye et al. [43] proposed on-demand situation-ware multicast where a node dynamically maintains a multicast tree using the topology information gathered from underlying unicast routing protocols. Abdulla et al. [1] used various DTN routing protocols to support DTN multicast such as Spray and Wait [38]. Chuah et al. [4] proposed Encounter-Based Multicast Routing (EBMR) that uses the encounter history based on PROPHET unicast DTN routing [29]; i.e., a node disseminates a packet to neighbors each of which has the highest delivery to one of the multicast receivers. Message ferrying was also used to support DTN multicast where DTN routing is aided by nodes whose mobility patterns are known, or whose trajectories are controllable [47], [42].

Throughput and Delay Scaling Behavior in DTNs: It is known that DTN routing protocols can benefit from node mobility and overcome the capacity bound of $\Theta(1/\sqrt{n \log n})$ originally established by Gupta and Kumar [16] for a fixed wireless network. Noting that the average hop length of a path is the key limiting factor, Grossglauser and Tse proposed the two-hop relay routing algorithm that exploits node mobility to effectively reduce the hop length, and utilizes relay nodes to deliver data to the destination when they meet, thus achieving $\Theta(1)$ throughput per node [15]. Various mobility models have been considered to characterize the delay/capacity relationship with respect to node mobility, from a simple independent and identically distributed (I.I.D.) mobility model [30], to more complex random mobility models, such as random waypoint [35], random direction [34], Brownian mobility [28], and random walk [11]. Sharma et al. systematically studied the impact of different mobility models on delay/capacity tradeoffs [34]. Garetto et al. studied a home-point mobility model where each node moves around its home-point, and studied its impact on capacity scaling properties [12]. In addition, the impact of finite buffer constraints on each node to the capacity of network has also been studied [17]. Note that besides the scaling behavior analysis, there is a body of work on the performance analysis of DTN routing protocols. Groenevelt et al. analyzed the average latency and the number of transmissions for the two-hop relaying and unrestricted replication (i.e., epidemic dissemination) using a stochastic model [14]. Hanbali et al. studied the maximum relay throughput and buffer occupancy of the two-hop relay routing under various mobility patterns [2]. Similarly, Zhang et al. studied on the performance of epidemic routing and its variations using a simple deterministic model [45].

VII. Conclusion

We investigated the throughput and delay scaling properties of multicasting in DTNs. We analyzed the maximum throughput bound of DTN multicast. We then proposed RelayCast, a routing scheme that extends the Grossglauser and Tse's two-hop relay algorithm, and showed that RelayCast achieves the maximum throughput of $\Theta(\min(n\lambda, \frac{n^2\lambda}{n_s n_d}))$ where n_s is the number of sources and n_d is the number of receivers associated with each source. We compared throughput and delay properties of RelayCast with those of conventional wireless multicast schemes and showed that RelayCast is much more scalable. We analyzed the impact of various network parameters and routing strategies on the throughput and delay scaling properties of RelayCast, namely buffer size, multi-user diversity among multicast receivers, and throughput-delay tradeoffs. In particular, we found that: (1) given finite buffer of size K, the throughput is reduced to $\Theta(\frac{K\lambda}{n_d})$, (2) multicast receiver relay where multicast receivers cooperate in delivering the packets can be exploited only if the number of multicast receivers scales as $\Theta(n)$, and (3) throughput can be traded for delay using RelayCast with k-copy replication. Finally, we validated our findings via extensive simulations.

REFERENCES

[1] M. Abdulla and R. Simon. A Simulation Analysis of Multicasting in Delay Tolerant Networks. In *WSC'06*, Monterey, CA, Dec. 2006.

[2] A. Al-Hanbali, A. A. Kherani, R. Groenevelt, P. Nain, and E. Altman. Impact of Mobility on the Performance of Relaying in Ad Hoc Networks. In *INFOCOM'06*, Barcelona, Spain, Apr. 2006.

[3] H. Cai and D. Y. Eun. Crossing Over the Bounded Domain: From Exponential To Power-law Inter-meeting Time in MANET. In *MobiCom'07*, Montréal, QC, Canada, Sept. 2007.

[4] M. C. Chuah and Y. Xi. An Encounter-Based Multicast Scheme for Disruption Tolerant Networks. Technical Report LU-CSE-07-009, Lehigh University, 2007.

[5] V. Conan, J. Leguay, and T. Friedman. Characterizing Pairwise Inter-contact Patterns in Delay Tolerant Networks. In *ACM Autonomics'07*, Rome, Italy, Oct. 2007.

[6] E. Daly and M. Haahr. Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs. In *MobiHoc'07*, Montréal, QC, Canada, Sept. 2007.

[7] R. M. de Moraes, H. R. Sadjadpour, and J. Garcia-Luna-Aceves. Throughput-Delay Analysis of Mobile Ad-hoc Networks with a Multi-Copy Relaying Strategy. In *SECON'04*, Santa Clara, CA, Oct. 2004.

[8] S. Deb, M. Médard, and C. Choute. Algebraic Gossip: A Network Coding Approach to Optimal Multiple Rumor Mongering. *IEEE Transactions on Information Theory*, 52(6), 2006.

[9] D.-Z. Du and F. K.-M. Hwang. A Proof of the Gilbert-Pollak Conjecture on the Steiner Ratio. *Algorithmica*, 7(2-3):121–135, 1992.

[10] C. Fernandess and D. Malkhi. On Collaborative Content Distribution using Multi-Message Gossip. In *IPDPS'06*, Rhodes, Greece, Apr. 2006.

[11] A. E. Gamal, J. Mammen, B. Prabhakar, and D. Shah. Throughput-Delay Trade-Off in Wireless Networks. In *INFOCOM'04*, Hong-Kong, Mar. 2004.

[12] M. Garetto, P. Giaccone, and E. Leonardi. Capacity Scaling in Delay Tolerant Networks with Heterogeneous Mobile Nodes. In *MobiHoc'07*, Quebec, Canada, Sept. 2007.

[13] R. Groenevelt. Stochastic Models in Mobile Ad Hoc Networks. Technical report, University of Nice, Sophia Antipolis, INRIA, 2006.

[14] R. Groenevelt, P. Nain, and G. Koole. The Message Delay in Mobile Ad Hoc Networks. In *Performance'05*, Juan-les-Pins, France, Oct. 2005.

[15] M. Grossglauser and D. N. C. Tse. Mobility Increases the Capacity of Ad Hoc Wireless Networks. *IEEE Transactions on Networking*, 10(4), Aug. 2002.

[16] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, 46(2), 2000.

[17] J. D. Herdtner and E. K. P. Chong. Throughput-Storage Tradeoff in Ad Hoc Networks. In *INFOCOM'05*, Miami, FL, Mar. 2005.

[18] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance Anomaly of 802.11b. In *INFOCOM'03*, San Francisco, CA, May 2003.

[19] S. Jain, M. Demmer, R. Patra, and K. Fall. Using redundancy to cope with failures in a Delay Tolerant Network. In *SIGCOMM'05*, Philadelphia, Pennsylvania, Aug. 2005.

[20] S. Jain, K. Fall, and R. Patra. Routing in a Delay Tolerant Networking. In *SIGCOMM'04*, Portland, OR, Aug.-Sep. 2004.

[21] T. Karagiannis, J.-Y. L. Boudec, and M. Vojnovic. Power Law and Exponential Decay of Inter Contact Times Between Mobile Devices. In *MobiCom'07*, Montréal, QC, Canada, Sept. 2007.

[22] A. Keshavarz-Haddad, V. Ribeiro, and R. Riedi. Broadcast Capacity in Multihop Wireless Networks. In *Mobicom'06*, Los Angeles, CA, Sept. 2006.

[23] L. Kleinrock. *Queueing Systems: Theory, Volume 1*. Wiley-Interscience, 1975.

[24] S. R. Kulkarni and P. Viswanath. A Deterministic Approach to Throughput Scaling in Wireless Networks. *IEEE Transactions on Information Theory*, 50(6):1041–1049, 2004.

[25] S.-J. Lee, M. Gerla, and C.-C. Chiang. On-demand Multicast Routing Protocol. In *WCNC'99*, New Orleans, LA, Sep. 1999.

[26] J. Leguay, T. Friedman, and V. Conan. Evaluating Mobility Pattern Space Routing for DTNs. In *INFOCOM'06*, Barcelona, Spain, Apr. 2006.

[27] X.-Y. Li, S.-J. Tang, and O. Frieder. The Multicast Capacity of Large Multihop Wireless Networks. In *MobiCom'07*, Montréal, QC, Canada, Sept. 2007.

[28] X. Lin, G. Sharma, R. R. Mazumdar, and N. B. Shroff. Degenerate Delay-Capacity Trade-offs in Ad Hoc Networks with Brownian Mobility. *IEEE/ACM Transactions on Networking (TON)*, 14(SI):2777–2784, 2006.

[29] A. Lindgren, A. Doria, and O. Schelén. PROPHET: Probabilistic Routing in Intermittently Connected Networks. In *SAPIR'04*, Fortaleza, Brazil, Aug. 2004.

[30] M. J. Neely and E. Modiano. Capacity and Delay Tradeoffs for Ad-Hoc Mobile Networks. *IEEE Transactions on Information Theory*, 46(2), June 2005.

[31] L. Qiu, Y. Zhang, F. Wang, M. K. Han, and R. Mahajan. A General Model of Wireless Interference. In *MobiCom'07*, Montréal, QC, Canada, Sept. 2007.

[32] S. Sanghavi, B. Hajek, and L. Massoulie. Gossiping With Multiple Messages. *IEEE Transactions on Information Theory*, 53(12):4640–4654, 2007.

[33] S. Shakkottai, X. Liu, and R. Srikant. The Multicast Capacity of Large Multihop Wireless Networks. In *MobiHoc'07*, Montréal, QC, Canada, Sept. 2007.

[34] G. Sharma, R. Mazumdar, and N. Shroff. Delay and Capacity Trade-offs in Mobile Ad Hoc Networks: A Global Perspective. In *INFOCOM'06*, Barcelona, Spain, Apr. 2006.

[35] G. Sharma and R. R. Mazumdar. Delay and Capacity Trade-off in Wireless Ad Hoc Networks with Random Way-point Mobility. Technical report, Purdue University, 2005.

[36] T. Small and Z. J. Haas. Resource and Performance Tradeoffs in Delay-Tolerant Wireless Networks. In *CHANTS'05*, Philadelphia, PA, Aug. 2005.

[37] T. Spyropoulos, K. Psounis, and C. Raghavendra. Single-Copy Routing in Intermittently Connected Mobile Networks. In *SECON'04*, Santa Clara, CA, Oct. 2004.

[38] T. Spyropoulos, K. Psounis, and C. Raghavendra. Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks. In *CHANTS'05*, Philadelphia, PA, Aug. 2005.

[39] B. Tavli. Broadcast Capacity of Wireless Networks. *IEEE Communication Letters*, 10(2), 2006.

[40] Y. Wang, S. Jain, M. Martonosi, and K. Fall. Erasure-Coding Based Routing for Opportunistic Networks. In *CHANTS'05*, Philadelphia, PA, Aug. 2005.

[41] Z. Wang, H. R. Sadjadpour, and J. Garcia-Luna-Aceves. A Unifying Perspective on the Capacity of Wireless Ad Hoc Networks. In *INFOCOM'08*, Phonex, AZ, Apr. 2008.

[42] P. Yang and M. C. Chuah. Context-Aware Multicast Routing Scheme for Disruption Tolerant. In *International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN'06)*, Malaga, Spain, Mar. 2006.

[43] Q. Ye, L. Cheng, M. C. Chuah, and B. D. Davison. OS-multicast: On-demand Situation-aware Multicasting in Disruption Tolerant Networks. In *VTC'06*, Melbourne, Australia, May 2006.

[44] L. Ying, S. Yang, and R. Srikant. Coding Achieves the Optimal Delay-Throughput Tradeoff in Mobile Ad Hoc Networks: Two-Dimensional I.I.D. Mobility Model with Fast Mobiles. In *WiOpt'07*, Limassol, Cyprus, Apr. 2007.

[45] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance Modeling of Epidemic Routing. In *IFIP Networking'06*, Coimbra, Portugal, May 2006.

[46] W. Zhao, M. Ammar, and E. Zegura. Multicasting in Delay Tolerant Networks: Semantic Models and Routing Algorithms. In *CHANTS'05*, Philadelphia, PA, Aug. 2005.

[47] W. Zhao, M. Ammar, and E. Zegura. Multicasting in Sparse MANETs using Message Ferrying. In *WCNC'06*, Las Vegas, NV, Apr. 2006.