# Helper Node Allocation Strategies for Content Dissemination in Intermittently Connected Mobile Networks

*Abstract*—**We formulate and address mathematically the fundamental problem of resource allocation in the form of helper nodes in disseminating multiple content in a hybrid intermittently connected mobile network under a general stochastic homogeneous contact process. We consider and solve two variations of the problem - one in which the goal is to maximize the expected demands satisfied and another in which the goal is to minimize the time taken to disseminate the contents. Besides the global optimization perspective, we also examine the problem from a game theoretic perspective in which a central agent auctions the storage to competing content providers, and show how self-interested decisions impact the social welfare.**

## I. Introduction

With the increasing availability of Wifi-direct equipped mobile devices and the planned introduction of wireless access for vehicular environments (WAVE) radios in vehicles in the near future, there is continuing interest in mobile applications involving encounter-based intermittently connected mobile networks (ICMN), that can off-load the dissemination of delay-tolerant content from the increasingly congested and expensive cellular data infrastructure.

Hitherto most of the literature on ICMN's (also referred to as delay-tolerant networks (DTN)) has focused primarily on a flat decentralized architecture. We argue that in the real-world, whether deployed on personal mobile devices or in vehicles, the omnipresent availability of cellular infrastructure makes possible an alternative, hybrid two-tier ICMN architecture. Analogous to the OpenFlow architecture [14], such a hybrid network would have separate control and data planes. The control plane would provide low-overhead bidirectional control messaging between devices and a server via the cellular infrastructure that allows for centralized resource allocation. The data plane would be where heavier amounts of data are disseminated via the encounter-based ICMN.

The focus of this work is on optimizing multicast dissemination of multiple contents in such a hybrid network. Specifically, we consider a central agent who has control over the storage resources of a set of helper relay nodes in such a hybrid two-tier network. A number of content providers are interested in disseminating (multicasting) the file they each have from a set of initial seed nodes to a set of demanding nodes for each content in the network.

The fundamental question we explore in this work through mathematical modeling under a general stochastic homogeneous encounter model is *how should the agent most efficiently allocate the helper nodes?* We consider this question under two different metrics — one that aims to maximize the number of satisfied demands by a deadline, and another that aims to minimize the time taken to satisfy all demands. And we explore this question under two settings. Initially, we consider a social welfare model where the agent is trying to maximize the total utility for all content providers. While this is the typical "engineering" approach to the problem, in the real world, the content provider and the resource managing agent are often different entities with differing interests. Therefore, we consider an economic model where the agent is assumed to be self-interested and trying to maximize its revenue from all content providers, which in turn have to balance their gain in terms of the content dissemination utility with the payment they must make to the agent. Our game-theoretic model allows us to examine the price of anarchy, the ratio between the total utility achieved when the system is operated based on maximizing revenue versus when the resources are allocated to maximize social welfare.

The high delays and reliability issues associated with single-copy routing in ICMN has motivated several researchers to develop multi-copy dissemination approaches in which the number of nodes assisting in relaying each content is either limited statically or carefully adapted [22], [1], [3]. In terms of the resource that is limited, many of these works emphasize the bandwidth limitation of ICMN [22], [23], [1], but several others have emphasized the storage limitation [18], [17], [20], [13], as we also do in this work.

Our work is closest in spirit to RAPID [3], whose authors make the important observation that efficient dissemination in ICMN should fundamentally be formulated as a utility-based resource allocation problem of how the limited storage resources on the nodes should be managed towards maximizing well-defined system objectives. They rightly distinguish their work from prior literature as being the first where the resource management has an *intentional* effect on desired performance metrics related to average delay or delivery rate as opposed to prior schemes that have only an *incidental* effect on these metrics. Our work closely follows this top-down philosophy, and we thus start by clearly identifying the relevant performance metrics and then derive the optimal storage allocation to maximize those metrics. However, our work is distinguished from [3] in that they propose heuristic mechanisms and evaluate them via simulations, while our objective here is to undertake a theoretical treatment of the problem in order to study rigorously the nature of optimal allocations under a well-defined, tractable, mathematical model.

We derive mathematical expressions for computing the expected time to satisfy all demands and the expected number of satisfied demands by a given deadline for a given helper node storage allocation, under a homogenous stochastic encounter model with general inter-encounter time distribution. Using this, we show how to compute the node allocation that maximizes the social welfare under both metrics. We show some interesting trends: for instance, helper nodes have diminishing returns and are less effective at large deadlines, and that increase in demand is actually beneficial in reducing the expected delay in dissemination. We formulate the problem of helper node allocation also from a game theoretic perspective and show that when the central agent tries to maximize its profit under a proportional allocation policy, the resulting system generally has a price of anarchy greater than 1. We also find that, somewhat counter-intuitively, a content provider with lower demand may need to pay more to the agent.

## II. Related Work

There have been a number of works that have focused on a mathematical treatment of content dissemination in DTN. We briefly survey these here.

One of the first works to analyze message dissemination for DTNs is [6]. It analyzes the expected delay in propagating a single message in a DTN using Markov models. We follow a similar approach of using Markov models, but our work is a more general version of their's - since we consider dissemination of multiple files simultaneously (multi-content) and each possibly to multiple nodes (multicast). Furthermore, we also analyze the expected number of demands satisfied by a deadline - a metric that might be useful for certain types of content.

The work [6] lead to [26], where the authors characterize the message delay approximately using ODEs. While we could have used the ODE approximations rather than the Markov model, we chose the latter because of the more general nature of the Markov model.

In [21], the authors are primarily concerned with analyzing the expected node inter-encounter durations for various mobility models, and for a more realistic mobility model that they derive. In [11], the authors derive optimal policies for buffer management because the amount of buffer and bandwidth is limited. While the authors make use of a similar contact model as well as similar metrics, their goal however is to decide when to drop items from the buffer.

In [8], the problem of disseminating news and other dynamic content to a mobile phone based ICMN is considered. It is shown how to determine an optimal allocation of the bandwidth of the service provider to maximize the social welfare of the network, such that the content at the users is as fresh as possible. [2] has a similar flavor, but the goal is to design efficient ways for distributing dynamic content when the participating nodes could be either cooperative or non-cooperative.

| Notation | Meaning |
|---|---|
| $N$ | number of nodes in the system. |
| $m$ | Number of files/number of content providers. |
| $[k]$ | Equivalent to $\{1, \dots, k\}$ for integer $k$. |
| $n_{d,i}, n_{h,i}$ | Number of demands and number of helper nodes allocated for file $i \in [m]$ |
| $c_{d,i}, c_{h,i}$ | Number of completed demands and number of completed helper nodes for file $i \in [m]$ with $0 \leq c_{d,i} \leq n_{d,i}$ and $0 \leq c_{h,i} \leq n_{h,i}$. |
| $\{X_j\}_1^\infty$ | Random variables denoting the inter-encounter times. |
| $U_i$ | Utility for content provider $i \in [m]$. |
| $A$ | The transition matrix of content dissemination for a fixed file of size $n_S \times n_S$. |
| $g_i^\tau(c_d, c_h)$ | The probability that $c_d$ of the demanding nodes and $c_h$ of the helper nodes (and not any more) have got the file at encounter $\tau$ for file $i \in [m]$. |

TABLE I: List of variables used.

[15] and [9] are two works that consider optimal distributed cache allocations strategies in ICMNs, along the lines of the well-known square root replication scheme [4]. While [15] is optimal at equilibrium, the convergence is not guaranteed; [9] overcomes this issue. Even though [15], [9] and our work share the similar objective (among others) of reducing access delay to content, their problem formulation and the setting is fundamentally different from ours.

Recently, DTNs have been started to be analyzed from a game theoretic setting. The primary argument here is that the helper (relay) nodes may be inherently selfish and so may not be willing to relay content for other nodes. Some works consider fully selfish nodes [25], [19], whereas others consider socially selfish nodes [12] - where nodes may be willing to carry content of other nodes depending on the social ties (for example, nodes may prefer to help friends rather than strangers). In our work, we do not consider fully autonomous nodes, but assume that a central agent can control the nodes. We consider instead self-interested content providers and a revenue-maximizing central agent.

## III. Problem Formulation

The model we consider here has a central agent that has control over allocating storage on a set of $N$ nodes, and $m$ content providers with one file each. Each content provider is interested in disseminating its file to some of the nodes that may be interested in the content. All the nodes are assumed to be homogeneous and are identical in their storage capabilities.

The central agent can get content (file) from the content providers and can place them in one or more nodes, called the seed nodes of the file (assume this is done offline). Let the number of seed nodes for each file $i$ be $n_{s,i}$ with $n_{s,i} \geq 1$. We will assume for simplicity that the central agent places each file onto different seed nodes. This is because we will be able to decouple the analysis of the dissemination of different files. Even if we assume that seed nodes store multiple files, all the analysis should be extendable with suitable bookkeeping.

There are a set of demanding nodes for each file. Assume that the number of demands for each file $i$ is $n_{d,i}$. The goal is to get the file $i$ to the $n_{d,i}$ demanding nodes.

Since the total number of nodes is $N$, we would require $\sum_{i=1}^{m}(n_{s,i}+n_{d,i}) \leq N$. Denote by $n_d = \sum_{i=1}^{m} n_{d,i}$ the total number of demand nodes, $n_s = \sum_{i=1}^{m} n_{s,i}$ the total number of seed nodes.

The remaining set of nodes, $N-n_s-n_d$ in number are called helper nodes as they can potentially help in the dissemination of the files. Thus the number of helper nodes is $n_h = N - n_s - n_d$. We assume that each of these helper nodes help in the dissemination of a single file (even though their storage capacities may be much higher). The central agent uses the control-plane prior to the dissemination process to inform each helper node which file it should be helping. When two nodes encounter, it is assumed that each node can download a full file, if the other node carries anything of interest. The encounter model will be made clear later.

It is of interest to determine which set of helper nodes must assist in the dissemination of each file and for the metric over which we want to optimize for, we consider the following two formulations:

- Metric 1 (M1): Maximize the total expected number of demands satisfied for each file, by a deadline $T$.
- Metric 2 (M2): Minimize the maximum expected time to satisfy all the demands for each file.

The metric to be used depends on the application. For example, M1 might be a useful metric when the content will expire after certain duration.

### A. Notation

The indices start with 1. All the vectors are column-vectors unless specified otherwise. $(A)_{(x,y)}$ refers to the element at row $x$ and column $y$ of matrix $A$, and $(\mathbf{v})_x$ refers to the element at location $x$ of the vector $\mathbf{v}$. Note that all the vectors considered here are of appropriate sizes, *i.e.* when a matrix $B$ multiplies a vector $\mathbf{v}$, the number of columns of $B$ equals the number of rows of $\mathbf{v}$. $[m]$ denotes $\{1, 2, \ldots, m\}$ for an integer $m$. A list of the symbols used in this paper are presented in Table I.

### B. Contact and Dissemination Model

We assume that content dissemination occurs through a series of encounters between nodes. In particular, we assume that when there is an encounter between nodes, it is always between a pair of nodes and not any more. Furthermore, we assume a homogeneous contact process whereby each pair of nodes are equally likely to be involved in an encounter. Since there are $N(N-1)/2$ possible pairs, in each encounter, each particular pair is involved with probability $p \triangleq \frac{2}{N(N-1)}$.

The inter-encounter times are assumed to be independent and identically distributed (i.i.d). Let $\{X_j\}_1^\infty$ be the i.i.d inter-encounter times, with mean $E[X]$. Thus, the expected time for say $\tau$ encounters is $\tau E[X]$. Let $F_X(t)$ be the cdf of $X$, i.e. $F_X(t) = \Pr[X \leq t]$. We would also like to define the joint distribution of time taken by $\tau$ encounters as follows: $f_\tau(t) = \Pr[\sum_{j=1}^{\tau} X_j \leq t]$.

These quantities depend on the underlying mobility. For example, if the inter-encounter times are i.i.d exponential with rate $\lambda$, then $F_X(t) = 1 - \exp(-\lambda t)$ and $f_\tau(t)$ is Erlang with parameters $\lambda$ and $\tau$ as follows: $f_\tau(t) = \frac{\lambda^\tau t^{\tau-1} e^{-\lambda t}}{(\tau-1)!}$.

We note that our contact model captures the widely used model in the DTN community. For example, [6], [26], [20], [11], all assume that node encounter times between pairs of nodes are independent of each other and follow a Poisson process with rate $1/\lambda$. This means that the inter-encounter durations are i.i.d exponential with rate $\lambda$, and thus each pair is equally likely to encounter, whenever there is one. Our analysis does not rely on the exponential assumption, and can capture a general inter-encounter distribution.

Since the state of the system changes only during an encounter, when presenting results, we will deal with the counting of the number of encounters instead of the time.

Finally, as mentioned before, it is assumed that the encounters are long enough to transmit a full file. As our primary focus is on large files and constrained storage, one way to satisfy this is to have large bandwidth. To handle settings in which both bandwidth and storage are constrained, this assumption is equivalent to assuming that only long-duration encounters are explicitly considered in our model. These long-duration encounters would typically correspond to individuals that are standing/sitting near each other, or to vehicles either temporarily parked near each other or following each other in the same direction[1]. In principle, the model could be extended in an approximate way to consider even partial file transmissions during short duration encounters by appropriately scaling down the probability of transferring a file in each encounter between a node that has the file and an interested node; we defer a careful exploration of such an approximation to future work.

## IV. Understanding Dissemination of a Single File

Let us first analyze the dissemination of a single file, with the assumption that all the nodes in the system are participating in the dissemination - *i.e.* they are either seeds, demands or are willing to help (helper nodes). If $n_s, n_d, n_h$ denote the number of seeds, demands and helpers, then we assume here that the total number of nodes $N = n_s + n_d + n_h$. Note that since there is only a single file, we do not consider notation like $n_{d,1}$ to denote the parameters for the first file.

We will utilize the solution of this section as the building block in the next, where we consider multiple files. There, we will also handle the case when there could be a single file with $n_s + n_d + n_h < N$. This can be handled by assuming that the remaining nodes participate in the dissemination of a non-existent file.

The dissemination of the file can be modeled using a finite-state discrete time absorbing Markov chain.

### A. Modeling the Markov Chain

*1) States of the Markov Chain:* The states of the Markov chain indicate how many demanding nodes and helper nodes

[1]Such long encounters are not rare. Prior work has shown that encounter durations for human contacts follow a heavy-tailed distribution in which long encounters are reasonably common [24], and that the contact durations for certain groups of vehicles are bimodal with long durations corresponding to the vehicles parked side by side [10].

have got the file. So a state can be represented using a tuple $(c_d, c_h)$ when $0 \leq c_d \leq n_d$ demands and $0 \leq c_h \leq n_h$ helpers have got the file. The state (0,0) for example indicates that none of the demands or helper nodes have got the file. The number of states is $n_S = (n_h + 1)(n_d + 1)$.

Note that since our goal is to be done with $n_d$ demands, all of the states $(n_d, j)$ could have been made absorbing states, or they could have been all combined to one absorbing state. We do not make this distinction when writing down the Markov chain because it lends to easier understanding of the analysis and it can capture the distribution of the number of helper nodes completed when all the demands are done.

*2) Transition Probabilities:* When in state $(c_d, c_h)$, the total number of nodes that have the file is $n_s + c_d + c_h$. The number of demands and helper nodes that don't have the file are thus $n_d - c_d$ and $n_h - c_h$ respectively. From a state $(c_d, c_h)$, the Markov chain can either go to state $(c_d + 1, c_h)$ or $(c_d, c_h + 1)$ or will stay in the same state. The corresponding transition probabilities are:

$$p_{(c_d,c_h)\to(c_d+1,c_h)} = p(n_s + c_d + c_h)(n_d - c_d) \qquad (1)$$

$$p_{(c_d,c_h)\to(c_d,c_h+1)} = p(n_s + c_d + c_h)(n_h - c_h) \qquad (2)$$

where $p = 2/N(N-1)$. The transition probability to the same state (self-loop) is

$p_{(c_d,c_h)\to(c_d,c_h)} = 1 - p_{(c_d,c_h)\to(c_d+1,c_h)} - p_{(c_d,c_h)\to(c_d,c_h+1)}$.

If the adjacent states $(c_d + 1, c_h)$ and/or $(c_d, c_h + 1)$ do not exist, set the corresponding transition probabilities to zero. Note that $(n_d, n_h)$ is an absorbing state.

*3) Transition matrix $A$:* Next, we construct the transition matrix $A$ which is of size $n_S \times n_S$ since $n_S = (n_h+1)(n_d+1)$ is the number of states. Note that the indexing of the rows and columns of the matrix start from 1. Each state of the Markov chain corresponds to a row or a column in the matrix at a particular index. Let $\text{In}(c_d, c_h)$ denote the index of state $(c_d, c_h)$ in the transition matrix $A$. We use the following mapping to map the state $(c_d, c_h)$ to an index in the matrix $A$: $\text{In}(c_d, c_h) = (n_h + 1)c_d + c_h + 1$.

Now given the index $i \in [n_S]$, $c_d = \text{quotient}(i - 1, n_h + 1)$ and $c_h = \text{remainder}(i - 1, n_h + 1)$ can give back the state $(c_d, c_h)$.

The transition probability from a state $(c_d, c_h)$ to $(c'_d, c'_h)$ will be stored in the location $(In(c_d, c_h), In(c'_d, c'_h))$ in the matrix.

The index of the absorbing state $(n_d, n_h)$ is $n_S$ and so $A$ can be written as:

$$A = \begin{bmatrix} A_0 & \mathbf{a} \\ \mathbf{0} & 1 \end{bmatrix}$$

where $\mathbf{0}$ is zero vector of appropriate size and $\mathbf{a}$ is a column vector. Let $\mathbf{1}$ represent the all ones vector of appropriate size. It can be proven by induction that for any $k$,

$$A^k = \begin{bmatrix} A_0^k & (I - A_0)^k \mathbf{1} \\ \mathbf{0} & 1 \end{bmatrix} \qquad (3)$$

*4) Occupancy of the Markov Chain:* Since the content dissemination starts at state $(0, 0)$, which corresponds to index $\text{In}(0, 0) = 1$, the initial probability distribution of being at various states is $\mathbf{e_1}$.

After $\tau$ encounters, the corresponding probability distribution is $\mathbf{e}_1^\intercal A^\tau$. Thus, the probability of being at state $(c_d, c_h)$ is the element in the index $i = \text{In}(c_d, c_h)$ of $\mathbf{e}_1^\intercal A^\tau$, which can also be represented as $(\mathbf{e}_1^\intercal A^\tau)_i$ and evaluates to $\mathbf{e}_1^\intercal A^\tau \mathbf{e}_i$.

Define $g^\tau(c_d, c_h)$ to be the probability that $c_d$ demanding nodes and $c_h$ helper nodes have got the file at the end of $\tau$ encounters. Then,

$$g^\tau(c_d, c_h) = \mathbf{e}_1^\intercal A^\tau \mathbf{e}_{\text{In}(c_d,c_h)}. \qquad (4)$$

### *B. A Few Definitions*

We will define three column vectors $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$ here. Let $\mathbf{u}_1$ be a column vector of length $n_S$ defined as follows:

$$\mathbf{u}_1 \triangleq \sum_{c_d=0}^{n_d} \sum_{c_h=0}^{n_h} c_d \mathbf{e}_{\text{In}(c_d,c_h)},$$

where $\mathbf{e}_i$ is a column vector of length $n_S$ with 1 at location $i$ and all other locations set to 0. The right hand side is also equivalent to $\sum_{i=0}^{n_S} \text{quotient}(i - 1, n_h + 1)e_i$, *i.e.* $\mathbf{u}_1$ has the first $n_h + 1$ entries 0, the second $n_h + 1$ entries 1, and so on and up to the last $n_h + 1$ entries set to $n_d$.

Let $\mathbf{u}_2$ be a column vector of length $n_S$ defined as follows:

$$\mathbf{u}_2 \triangleq \sum_{c_h=0}^{n_h} p(n_s + n_d - 1 + c_h)\mathbf{e}_{\text{In}(n_d-1,c_h)} \qquad (5)$$

Note that $(\mathbf{u}_2)_{n_S} = 0$.

Furthermore, define $\mathbf{u}_3$ to be a column vector of length $n_S - 1$, containing the first $n_S - 1$ entries of $\mathbf{u}_2$ as follows:

$$(\mathbf{u}_3)_i \triangleq (\mathbf{u}_2)_i, \qquad \forall i = 1, 2, \ldots, n_S - 1.$$

### *C. M1: Understanding the expected number of satisfied demands*

Here, we will derive an expression for the expected number of demands satisfied (also called completed demands) at the end of $\tau$ encounters. Then depending on the contact model, we will derive the same quantity, given a deadline. We slightly abuse the notation to indicate $C_D(\tau)$ and $C_D(T)$ to be the number of completed demands after $\tau$ encounters and after time $T$ respectively.

*Lemma 1:* The expected number of completed demands after $\tau$ encounters is

$$E[C_D(\tau)] = \mathbf{e}_1^\intercal A^\tau \mathbf{u_1}. \qquad (6)$$

*Proof:* At the end of $\tau$ encounters, the probability distribution in being at any of the states is given by $g^\tau(\cdot, \cdot)$. When in state $(c_d, c_h)$, the number of demands satisfied is $c_d$ and the corresponding probability is $g^\tau(c_d, c_h)$. Hence we have $E[C_D(\tau)] = \sum_{c_d=0}^{n_d} \sum_{c_h=0}^{n_h} c_d g^\tau(c_d, c_h)$. From equation 4, and then using the definition of $\mathbf{u}_1$, we will get the desired expression. ∎

*Theorem 1:* Given a deadline $T$, the expected number of demands satisfied by a deadline $T$ is

$$E[C_D(T)] = \int_0^T [1 - F_X(T-t)]\, \mathbf{e}_1^\intercal \sum_{\tau=0}^{\infty} A^\tau f_\tau(t)\mathbf{u}_1 dt \tag{7}$$

*Proof:* $E[C_D(T)] = \sum_{\tau=0}^{\infty} E[C_D(\tau)] \Pr\left[\sum_{j=1}^{\tau} X_j \leq T < \sum_{j=1}^{\tau+1}\right]$. Conditioning on duration of $\tau$ encounters and noting that $X_j$ are independent, we can write $\Pr\left[\sum_{j=1}^{\tau} X_j \leq T < \sum_{j=1}^{\tau+1}\right] = \int_0^T \Pr\left[X_{\tau+1} > T - t\right] f_\tau(t)dt = \int_0^T [1 - F_X(T-t)]\, f_\tau(t)dt$. Thus, $E[C_D(T)] = \int_0^T [1 - F_X(T-t)] \sum_{\tau=0}^{\infty} E[C_D(\tau)] f_\tau(t)dt$. Using equation 6 from Lemma 1 completes the proof. ■

### *D. M2: Understanding the expected completion Time*

*Theorem 2:* Given the transition matrix $A$ for the dissemination of a file, the expected delay (in units of time) to disseminate the file to all $n_d$ demands is

$$E[D] = \mathbf{e}_1^\intercal (I - A_0)^{-2}\mathbf{u}_3 E[X] \tag{8}$$

where $A_0$ is the $(n_S - 1) \times (n_S - 1)$ submatrix of $A$, $\mathbf{e}$ is a unit vector $[1, 0, \ldots, 0]^\intercal$ of size $n_S - 1$ and $\mathbf{u}_3$ is defined above.

*Proof:* We will first count the expected number of encounters to satisfy all demands and then multiply it by $E[X]$ to get $E[D]$. If the last encounter that satisfies all demands is $\tau$, then $d-1$ demands must have been satisfied at the end of $\tau - 1$ encounters. $E[D] = E[X] \sum_{\tau=1}^{\infty} \tau \Pr[C_D(\tau) = n_d \mid C_D(\tau-1) = n_d - 1]$. Also, by conditioning on the number of helper nodes $C_H$ done at the end of $\tau - 1$ encounters,
$\Pr[C_D(\tau) = n_d \mid C_D(\tau - 1) = n_d - 1]$

$$= \sum_{c_h=0}^{n_h} \Pr[C_D(\tau) = n_d \mid C_D(\tau-1) = n_d - 1, C_H(\tau-1) = c_h] \Pr[C_D(\tau-1) = n_d - 1, C_H(\tau - 1) = c_h]$$

$$= \sum_{c_h=0}^{n_h} \Pr[C_D(\tau) = n_d \mid C_D(\tau-1) = n_d - 1, C_H(\tau-1) = c_h] g^{\tau-1}(n_d - 1, c_h).$$

where $C_H(\tau)$ is the random variable indicating the number of helper nodes done at the end of $\tau$ encounters.

The probability term here is just the transition probability $p(n_s + n_d - 1 + c_h)$. Using this, and then using equation 4, we get $E[D] = E[X] \sum_{\tau=1}^{\infty} \tau \mathbf{e}_1^\intercal A^{\tau-1}\mathbf{u}_2$. We cannot write $\sum_{\tau=1}^{\infty} \tau A^{\tau-1} = (\mathbf{I} - A)^{-2}$ since $\mathbf{I} - A$ is singular. But since the last entry of $\mathbf{u}_2$ is 0, post-multiplying $\mathbf{e}_1^\intercal A^{\tau-1}$ by $\mathbf{u}_2$ is equivalent to $\mathbf{e}_1^\intercal A_0^{\tau-1}\mathbf{u}_3$ (where $\mathbf{e}_1$ is of length $n_S - 1$). Applying taylor series to the infinite sum completes the proof. ■

In Fig 1 we show the expected number of demands satisfied as a function of the number of encounters so far (or they can be considered as the deadline) obtained using equation 6. The expected number of encounters to satisfy all the demands is found using equation 8. There are $N = 50$ nodes, $n_s = 1$ seed, $n_d = 30$ demands and the remaining are helper nodes. We also utilized a custom built simulator with the specified node mobilities. The average number of encounters to satisfy all demands as per the simulation was 206.47 and it can be seen that the results obtained closely match the theoretical results.

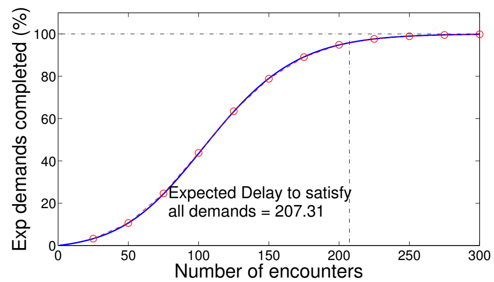



Fig. 1: The expected number of demands satisfied in percentage as a function of the number of encounters.

## V. Understanding Dissemination of multiple files

Not only do we generalize to multiple file dissemination here, but we could also generalize to the case where the number of seeds, demands and helper nodes do not have to add up to the total number of nodes in the system.

We can consider all the nodes associated with each file $i$ as a group, such that the number of nodes in group-$i$ are $N_i = n_{s,i} + n_{h,i} + n_{d,i}$. Also, $\sum_{i=1}^{m} N_i = N$.

The general idea is to apply the previous results of M1 or M2 for each of the groups as if only the dissemination of nodes in the groups is going on, and then combine the results carefully to obtain the results of M1 or M2 when considering all the files together.

### *A. M1: Compute Expected Number of Satisfied Demands*

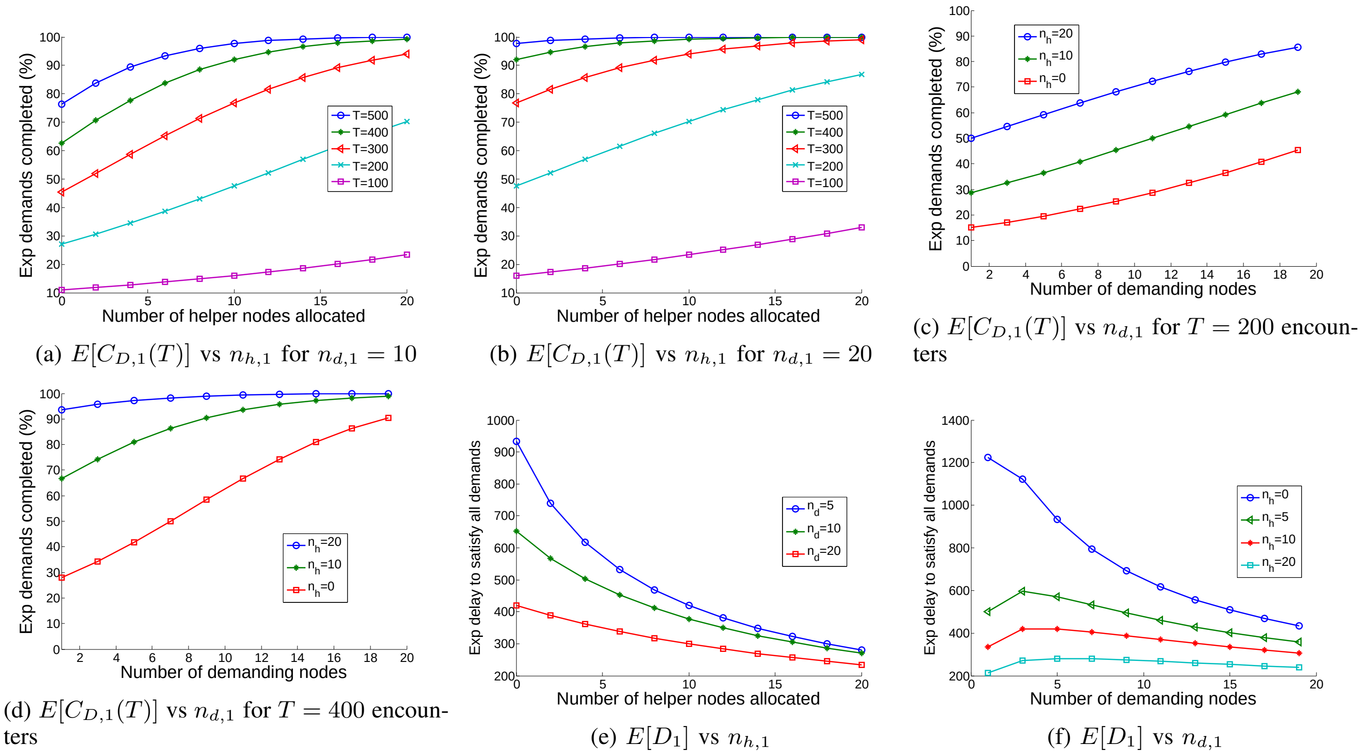
Given a deadline $T$ (in units of time), we want to compute the expected number of satisfied demands $E[C_D(T)]$. This can be expressed in terms of $E[C_D(\tau)]$, the expected number of demands satisfied after $\tau$ encounters in the system. If $E[C_D(\tau)]$ is known, by the same approach as outlined in Theorem 1 we can get

$$E[C_D(T)] = \int_0^T [1 - F_X(T-t)] \sum_{\tau=0}^{\infty} E[C_D(\tau)] f_\tau(t)dt. \tag{9}$$

Let us next compute $E[C_D(\tau)]$.

When there was a single file before, after $\tau$ encounters in the system, the corresponding Markov chain would have taken $\tau$ steps. But when there are $m$ files, we will have $m$ Markov chains that are not independent of each other.

Given an encounter in the system, the probability that the encounter is useful for file $i$ is $p_i = \frac{N_i(N_i-1)}{N(N-1)}$ (when nodes within this group encounter). This is the probability that the




(a) $E[C_{D,1}(T)]$ vs $n_{h,1}$ for $n_{d,1} = 10$

(b) $E[C_{D,1}(T)]$ vs $n_{h,1}$ for $n_{d,1} = 20$

(c) $E[C_{D,1}(T)]$ vs $n_{d,1}$ for $T = 200$ encounters

(d) $E[C_{D,1}(T)]$ vs $n_{d,1}$ for $T = 400$ encounters

(e) $E[D_1]$ vs $n_{h,1}$

(f) $E[D_1]$ vs $n_{d,1}$

Fig. 2: We consider $N = 50$ nodes that could have multiple files, and show various statistics for file 1.

Markov chain corresponding to file $i$ will make a move when there is an encounter in the system.

If we know that the number of encounters pertaining to group-$i$ is $\tau_i$, the results from the previous section can help us determine the expected number of demands satisfied $E[C_{D,i}(\tau_i)]$ (use Lemma 1 with appropriate parameters, e.g. the total number of nodes will be $N_i$ etc.).

Thus, conditioned on the number of encounters that could have helped in the dissemination of file $i$, the expected number of demands satisfied for file $i$ at the end of $\tau$ encounters in the system will be $\sum_{\tau_i=0}^{\tau} \binom{\tau}{\tau_i} p_i^{\tau_i}(1-p_i)^{\tau-\tau_i} E[C_{D,i}(\tau_i)]$.

Since $C_D(\tau)$ is just the sum of the number of demands satisfied for each file, by linearity of expectations, $E[C_D(\tau)] = \sum_{i=1}^{m} \sum_{\tau_i=0}^{\tau} \binom{\tau}{\tau_i} p_i^{\tau_i}(1-p_i)^{\tau-\tau_i} E[C_{D,i}(\tau_i)]$ Once $E[C_D(\tau)]$ is computed, and then using equation 9, we can compute the expected number of demands satisfied by deadline $T$, $E[C_D(T)]$.

After $\tau$ encounters, the expected number of demands satisfied for file $i$ is $E[C_{D,i}(\tau)]$, and so similar to equation 9, given a deadline $T$, the corresponding expectation will be

$$E[C_{D,i}(T)] = \int_0^T [1 - F_X(T-t)] \times \sum_{\tau=0}^{\infty} \sum_{\tau_i=0}^{\tau} \binom{\tau}{\tau_i} p_i^{\tau_i}(1-p_i)^{\tau-\tau_i} E[C_{D,i}(\tau_i)] f_\tau(t) dt \qquad (10)$$

### B. M2: Compute Completion Time

As in M1, we will divide the nodes into groups for each file and we can use Theorem 2 to compute the expected time ($E[D_i]$) and the expected number of encounters ($E[D_i]/E[X]$) within group-$i$ to satisfy all demands.

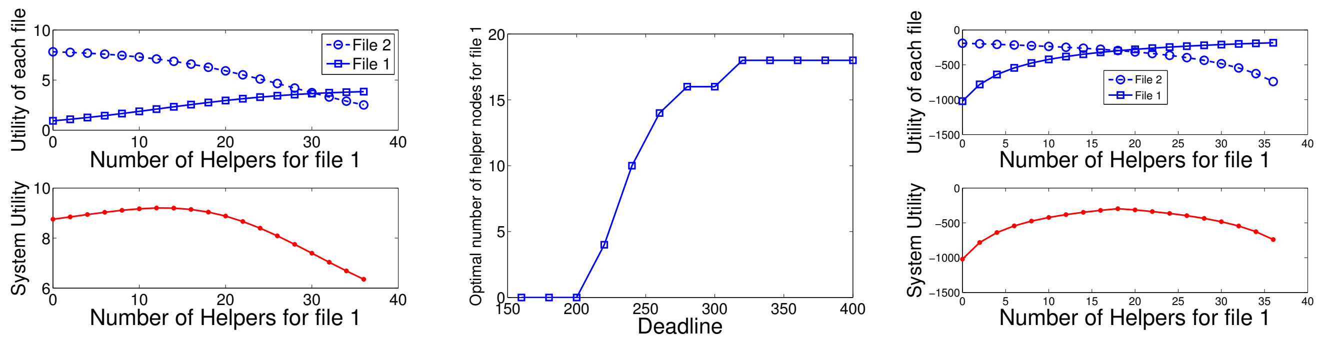
Since each encounter in the system will be a possible encounter within group-$i$ with probability $p_i = \frac{N_i(N_i-1)}{N(N-1)}$, the expected number of encounters in the system required to disseminate to all demands in group-$i$ will be $E[D_i]/E[X]p_i$ and the expected time will be $E[D_i]/p_i$.

We are interested in the maximum expected delay which will be $\max_{i\in[m]} E[D_i]/E[X]p_i$.

### C. Understanding M1 and M2

In Fig. 2, we show some characterizations of the dissemination under M1 and M2 for a single file. As previously mentioned, we only count the number of encounters rather than the actual time, which will depend on the distribution. We show results for say the first file. For M1, when deadline $T = \tau$, number of encounters in the system is given, $\sum_{\tau_1=0}^{\tau} \binom{\tau}{\tau_1} p_1^{\tau_1}(1-p_1)^{\tau-\tau_1} E[C_{D,1}(\tau_1)]$ captures the expected demands satisfied, whose percentage is plotted (obtained by dividing by $n_{d,1} = n_d$ and scaling by 100). For M2, $E[D_i]/E[X]p_i$ is plotted. For all the cases, we consider one seed.

Fig. 2a-Fig. 2d show plots when M1 is used, whereas the rest show plots when M2 is used. In Fig. 2a, we plot the percentage of demands that get satisfied at the end of various deadlines as a function of the helper nodes allocated when the total number




(a) M1: $n_{d,1} = 4$, $n_{d,1} = 8$. $U$ maximum when $n_{h,1} = 12$ and so $n_{h,2} = 24$. $T = 250$

(b) M1: Optimal number of helpers for file 1 for various deadlines

(c) M2: $n_{d,1} = 4$, $n_{d,2} = 8$. $U$ maximum when $n_{h,1} = 18$ and so $n_{h,2} = 18$

Fig. 3: Finding the globally optimal (social welfare maximizing) helper node allocation

of demands for the file is 10. As can be seen, the effect of the helper nodes is the highest for medium deadlines. When the deadline is small, even if a large number of helpers are allocated, the number of demands satisfied does not improve by the same numbers as when the deadline is medium. Further when the deadline is large enough, most of the demands are completed by virtue of the long deadlines, and the helpers offer only diminishing returns.

Fig. 2b shows a similar plot for 20 demands instead of 10. We can see that the curves have shifted upwards because having more demands actually helps the dissemination, as we will see next.

In Fig. 2c and Fig. 2d we fix the deadlines to $T = 200$ and $T = 400$ respectively, and study the percentage of demands completed in expectation as a function of the demands. These figures reveal that as the number of demands increase, the percentage completed also increase, even though this means that more demands have to be satisfied by the deadline. This happens because as the demands also increase, they can also contribute to the dissemination. If a few demands get the file before the deadline, they can still help other demands.

We next turn to studying M2 in Fig. 2e and Fig. 2f where we plot the expected delay (in number of encounters) as a function of either the number of helpers or the demands. In both the plots, it can be seen that increasing either the helpers or the demands can only help. From Fig. 2e it can be seen that when the demands are low, even a slight increase in the number of helpers allocated will bring down the expected delays significantly. Furthermore, it can be seen that the helper nodes have diminishing returns.

Fig. 2f offers an interesting view of the role of demands. As before, the overall trend here is that as the demands increase, the expected delay continues to decrease except at a few places. When the number of helper nodes is very low (or zero), any increase in the demands lends itself favorably, whereas when the number of helper nodes is higher, a slight increase in the demands registers itself as extra burden before starting to help itself.

## VI. Social Allocation

Having studied the dissemination of multiple files, we will now turn to finding an optimal allocation of the helper nodes to help in the dissemination of the files. In order to do so, depending on M1 or M2, we will define utilities for the dissemination of each of the files and then define the system utility. An allocation of the helper nodes defined as $(n_{h,1}, n_{h,2}, \dots, n_{h,m})$ indicates that $n_{h,i}$ helper nodes are allocated to help disseminate file $i$ and $\sum_{i=1}^{m} n_{h,i} = n_h$.

For M1, given a deadline $T$, set $U_i = E[C_{D,i}(T)]$ as the utility of the content provider $i$.

The system utility will be $U = \sum_{i=1}^{m} U_i$. and this is same as $E[C_D(T)]$. For M2, set $U_i = -\frac{E[D_i]}{p_i}$ as the utility of content provider $i$.

The utility of the system will be $U = \min U_i$.

Next, find an allocation $(n_{h,1}, n_{h,2}, \dots, n_{h,m})$ with $\sum_i n_{h,i} = n_h$, such that the system utility $U$ is maximized.

$$U_{\max} = \max_{(n_{h,1}, n_{h,2}, \dots, n_{h,m}), \sum n_{h,i} = n_h} \left( \max_i (U_i(n_{h,i})) \right).$$

In Fig 3, we show an example of how the allocation of helper nodes affects the utilities and how thereby we find the best allocation for M1 and M2. As before, we consider $N = 50$ nodes and two files with seeds $n_{s,1} = n_{s,2} = 1$ and demands $n_{d,1} = 4, n_{d,2} = 8$. Thus the number of remaining nodes are 36, which can be considered as the pool of helper nodes ($n_h = 36$). Since we are interested in allocating all the helper nodes to help either of the files, $n_{h,1} + n_{h,2} = 36$. In the plots, we vary $n_{h,1}$ and $n_{h,2}$ can be inferred.

From Fig 3a and Fig. 3c, the utilities of the individual files as well as the system utility are plotted as a function of the number of helper nodes allocated for file 1. Clearly, as $n_{h,1}$ increases, the utility of file 1 increases and that of the second file decreases because $n_{h,2}$ decreases. It can be seen that the utilities are maximized when $(n_{h,1}, n_{h,2}) = (12, 24)$ and $(18, 18)$ respectively.

Fig 3b shows an interesting effect of the deadline on the optimal allocation. We plot the optimal number of nodes allocated for the file 1 against various deadlines (in number of encounters). We note that when the deadline is very small, the helper nodes may not have much influence on file 1 anyway, so all of them are being allocated to file 2. But as the deadline increases, more and more are allocated until the optimal allocation stops at $(18, 18)$ for the nodes.

## VII. Market Allocation

To reiterate, there are $m$ content providers (CP), numbered $i = 1, 2, \ldots, m$. Content provider $i$ has a single file $i$, and is interested in disseminating the file to $n_{d,i}$ demanding nodes. Given a pool of helper nodes, so far we have studied the problem of allocating these helper nodes for efficient content dissemination from a social setting. Next, we would like to study the market-based scenario where the central agent requires the content providers to pay for helper nodes to help in the dissemination of their files[2]. Based on the bids placed by each content provider (CP), the central agent (CA) will allocate the sets of helper nodes to each of the CPs so as to maximize its own revenue.

Note that we could have a formulation where the CPs could each pay for the seed nodes in addition to the helper nodes (albeit at a higher cost). We exclude ourselves from doing that for ease of exposition. It is assumed that the central agent gets the files from the CPs and places each file in $m$ distinct nodes, which will form the seed nodes for the $m$ CPs. This allocation could be done by the use of control tier of the network, Thus $n_{s,i} = 1$ for all $i \in [m]$ irrespective of the bid placed by CP $i$.

### A. Game Formulation

*1) Players:* In our game formulation, we have $m+1$ players in total - the agent and the $m$ CPs with one file each. We will denote by player 0 the agent, and by player $i$, the CP $i$ with file $i$, for $i \in [m]$.

*2) Actions:* The action of each player is a price $c_i$. In the case of the agent, this price $c_0 > 0$ is the minimum unit price for the helper nodes that it requires the CPs to pay. The agent informs this value to all the other players.

The action of other players is then the price they want to bid $c_i \geq 0$ for $i \in [m]$.

Note that the agent moves first: it fixes the $c_0$ and informs the CPs. We thus have a *Stackelberg* game here. We assume that all the players have complete knowledge of the system. The agent thus knows all the demands for all the contents; each CP knows not only the demand of its respective content, but also of other contents. Furthermore each CP knows $N$ and the unit price $c_0$.

Since the players will not bid arbitrarily high values, we can restrict $c_i \leq c_{\max}$. Therefore, a row-vector $\mathbf{c} = [c_0, c_1, \ldots, c_m] \in [0, c_{\max}]^{m+1}$ forms the strategy space. Since $[0, c_{\max}]^{m+1}$ is a hypercube, it is a compact and convex subset of $\mathbb{R}^{m+1}$.

[2]We are assuming that in this two-tier network the storage in the nodes can be centrally controlled and managed by a single economic entity - the agent. There may be additional layer of economic interaction whereby this agent pays each individual node for each use of their storage. This payment can be absorbed into our model as a fixed cost for the central agent as it would still be interested in maximizing its revenue.

All the players will determine their actions to maximize their payoffs, which will be explained later.

*3) Allocation Policy:* The minimum price on each helper node is $c_0$. So if the CP $i$ pays $c_i$ and if $c_i < c_0$, then it will not get any helper nodes. If the price $c_i \geq c_0$, the player could possibly get $\lfloor c_i/c_0 \rfloor$, subject to the availability of the helper nodes and the price bid by other players.

In the case when the bid values are sufficiently high, such that $\sum_{i=1}^{m} \lfloor c_i/c_0 \rfloor$ is more than the number of helper nodes available, the agent could decide to allocate the helper nodes proportional to $c_i$. Thus, each CP could possibly get $\lfloor n_h c_i / (\sum_{j=1}^{m} c_j) \rfloor$. Combining both cases, the number of helper nodes allocated to CP $i$ as a function of the payment $c_i$ made by the CP and the payment of other players $c_{-i}$ can be expressed as follows:

$$n_{h,i}(c_i, c_{-i}) = \left\lfloor \min\left( \frac{c_i}{c_0}, \frac{n_h c_i}{\sum_{j=1}^{m} c_j} \right) \right\rfloor. \tag{11}$$

where $c_{-i} = (c_1, c_2, \ldots, c_{i-1}, c_{i+1}, \ldots, c_m)$.

*4) Payoffs:* The agent's utility is the sum of the prices accrued from each of the players minus the maintenance cost. since the maintenance cost is constant, it does not affect the equilibrium calculation and we omit it for simplicity. Thus, the payoff will be $P_0 = U_0 = \sum_{i=1}^{m} c_i$. As a Stackelberg-leader agent will set $c_0$ to maximize its payoff.

By bidding $c_i$, the CP $i$ gets $n_{h,i}$ helper nodes allocated to help in the dissemination of file $i$. Next, depending on M1 or M2, the CP will reap a utility $U_i(n_{h,i})$, which may be computed since we know the number of seeds $n_{s,i} = 1$, helpers $n_{h,i}$ and demands $n_{d,i}$.

Thus the net payoff of the CP $i \in [m]$ is

$$P_i = wU_i(n_{h,i}) - c_i, \tag{12}$$

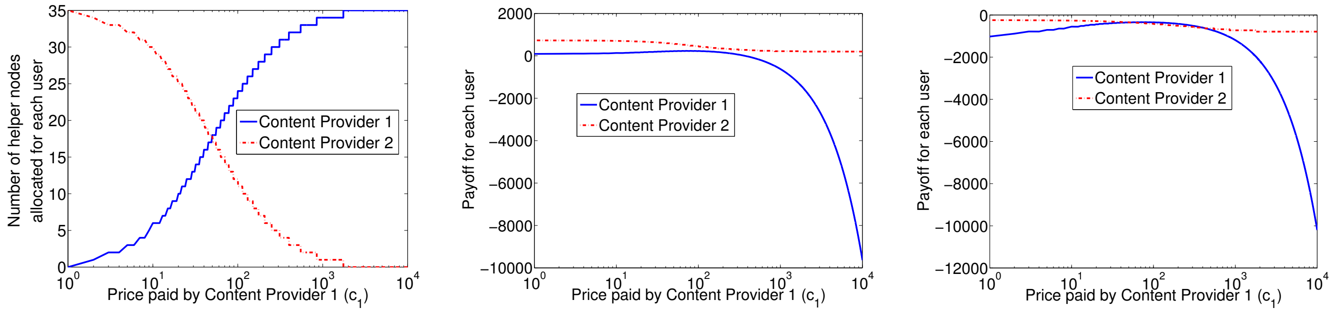
and it is of interest to the CP to maximize this. Here $w$ is a weighting parameter that dictates how much the CP values the outcome compared to the cost, and depends on the metric. It could vary from player to player, but we do not consider this distinction.

### B. Existence of Nash Equilibria

We numerically observe the existence of multiple Nash equilibria for the cases we consider (more details in the next subsection). We investigate the existence of Nash equilibria in a more detailed manner in Appendix A, where we rely on the quasiconcavity property of payoffs to find out cases where Nash equilibria are guaranteed to exist [16].

### C. Three Player Game Example

We next turn to numerically understand a three player game consisting of an agent and two CPs. Let there be $N = 50$ nodes in total of which $n_{d,1} = 4$ want content 1 and $n_{d,2} = 8$ want content 2. Irrespective of the bids placed by the content providers, the central agent will guarantee one seed each: $n_{s,1} = n_{s,2} = 1$. There are $n_h = 36$ helper nodes remaining. We need to pick $w$ for M1 and M2 to get the expressions for




(a) Number of helpers allocated to each content provider

(b) Utilities obtained by each content provider under M1 (Deadline = 250)

(c) Utilities obtained by each content provider under M2

Fig. 4: Three player example (with one agent and two content providers). Here $c_0 = 1$,$c_2 = 50$, $n_d(1) = 4, n_d(2) = 8$

payoff. Since the utility values from M2 are about two orders higher in magnitude than that in M1, we pick $w = 100$ for M1 while keeping $w = 1$ for M2. Given all this information, we are interested in determining what should the central agent fix $c_0$ to and once this is fixed, what will be the prices $(c_1, c_2)$ that the content providers bid?

Let us first understand the best response dynamics. If we fix $c_0$ to 1 and $c_2$ to 50. The number of helper nodes allocated according to equation 11 and the utilities of both the content providers for various values of $c_1$ is shown in Fig. 4. As can be seen, the number of helper nodes allocated increases as the content provider pays more and more, and it eventually gets almost all the helper nodes (the curve saturates at 35 rather than 36 because of the floor function in equation 11). Correspondingly, its payoff first increases since it is getting more helper nodes, but after a certain threshold, the payoff starts to fall due to the increasing cost. The best response of the content provider 1 will just be the price $c_1$ when the payoff is maximized. Under both M1 and M2, the best response of content provider 1 is to set $c_1 = 70$.

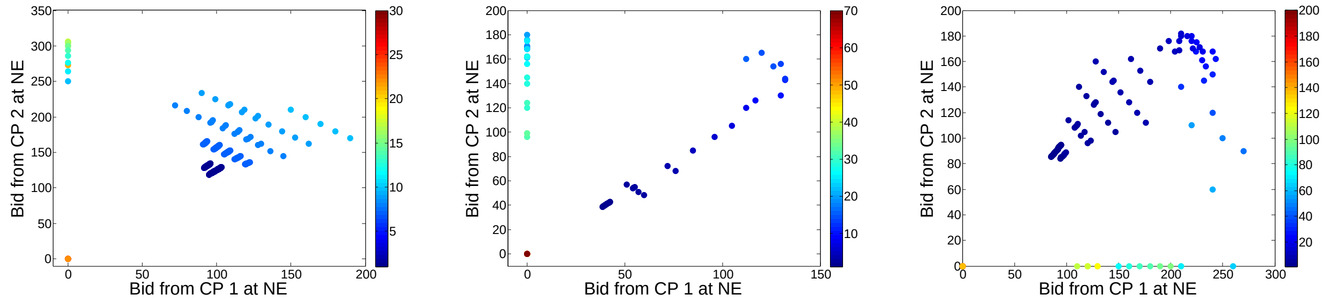
But since both the players know that they will play the best response to each other, they will play according to a Nash Equilibrium, if it exists. In fact, we numerically see that there are multiple pure Nash Equilibria when $c_0 = 1$. We show this in Fig. 5. In each figure, the circle represents a bid $(c_1, c_2)$ that could be a possible NE, and the circles are color coded according to $c_0$ (refer to the color bar legend to get approximate values of $c_0$). There are in fact many NE, but we do not show them all in the illustrations. The general trend to be noted is that as $c_0$ increases, the points shift slowly to the upper right side (which means the bids increase), but after a certain extent, one of the bidder realizes that it is too costly and so starts to bid zero (and so the other is non-zero). These are shown along the x-axis or y-axis. When the $c_0$ is too high, both the bidders bid 0 each. Another thing to note is that there are several equilibria when $c_0$ is low, but for higher values, there seems to be only one NE.

As the central agent increases $c_0$, it will start earning bigger payoffs until a certain extent depending on the scheme, after which the payoff starts to decrease and eventually reaches zero, see Fig. 6. To plot this curve, after fixing the model, for each $c_0$, we determine the possible Nash Equilibria $(c_1, c_2)$ numerically and use the one that gives the lowest $c_1 + c_2$.

For M1 with deadline 250, the payoff is maximized when $c_0 = 9$. The content providers bid $(90, 234)$ and get $(10, 26)$ helper nodes each. The expected demands satisfied for each of the content providers due to this allocation are 1.87 and 7.30, and thus the system utility is 9.17. Since the optimal allocation that maximizes the system utility is in fact $(12, 14)$, we should have that the system utility be $2.10 + 7.10 = 9.20$, where 2.10 and 7.10 are the expected number of demands satisfied for each at the end of 250 encounters with the help of 12 and 24 helper nodes. Thus, the price of anarchy is $9.20/9.17 = 1.0033$.

For M1 with deadline 400, the corresponding $c_0 = 13$. The bids here are $(130, 156)$ and the number of helper nodes allocated are $(10, 12)$. Note that all the helper nodes did not get allocated since the price minimum price is very high. Here, the utilities for the content providers are 3.1093 and 7.3683 respectively, and the overall system utility is 10.4776. The optimal allocation of helper nodes for this deadline is $(18, 18)$, which guarantees a system utility of $3.7895+7.8259 = 11.6154$. Thus, the price of anarchy is 1.1086, higher than before. Given the sufficiently long deadline, the improvement in utility brought by helper nodes for the second content provider is quite low as compared to the price it has to pay. Therefore, it will bid lower than it did when the deadline was shorter (when it knew that the helpers would indeed help).

In M2, the agent fixes $c_0$ to be 24 at which the bids will be $(240, 168)$. 10 and 7 helper nodes will be allocated for each content provider. In fact the content provider 2 bids much lower partly because of the much higher minimum price $c_0$ and partly because it knows that the demands can help themselves. The expected delay for content provider 1 is 422.3303 (and so its utility is -422.3303), and that of content provider 2 is 457.9848 (utility is -457.9848). The utility of the system is then -457.9848. If the allocation of the helper nodes were to maximize social welfare, the content providers would have got 18 helpers each, due to which the max delay would have been 297.2508 (each will have delay 297.2508 and 294.4246). Thus the price of anarchy in this case is 1.5407, the highest.




(a) M1 (Deadline = 250) (b) M1 (Deadline = 400) (c) M2

Fig. 5: The bids for different values of $c_0$.

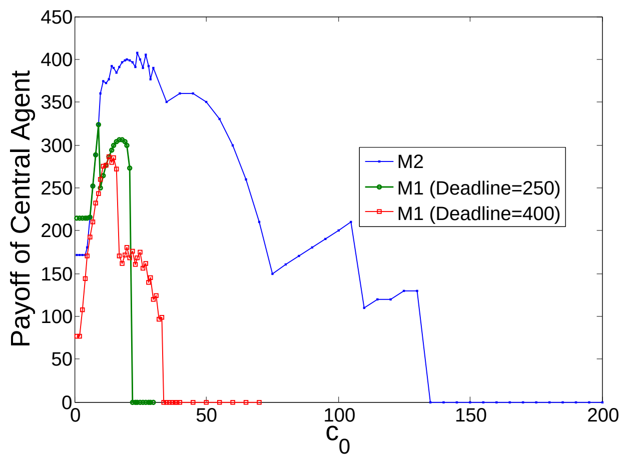



Fig. 6: Payoff of the Central Agent for various $c_0$ for M1 with deadlines 250 and 400 and M2. The Central Agent will fix the $c_0$ to maximize its payoff, and it depends on the scheme.

## VIII. Conclusion

We have formulated and analyzed the problem of helper node allocation in a hybrid ICMN mathematically under a general stochastic homogeneous encounter model. We believe this analysis advances our theoretical understanding of the impact of various parameters and designs (such as the impact of a market-based approach on social welfare) for resource allocation in ICMN.

Some recent works have explored heuristically the problem of allocating storage in the context of statistically structured heterogeneous ICMN, including via social network analysis [5], [7]. To shed some theoretical light on such problems, the formulation in this work will need to be extended in future work, to handle more realistic heterogenous mobility patterns. This may be mathematically and computationally challenging as not only the number of helper nodes, but also their identity starts to matter, resulting in a combinatorial explosion of states. Nevertheless, approaches leveraging approximation algorithms may prove fruitful.

## References


[1] M. Alresaini, M. Sathiamoorthy, B. Krishnamachari, and M.J. Neely. Backpressure with adaptive redundancy (bwar). In *INFOCOM 2012*.

[2] E. Altaian, P. Nain, and J.C. Bermond. Distributed storage management of evolving files in delay tolerant ad hoc networks. In *INFOCOM 2009*.

[3] A. Balasubramanian, B. Levine, and A. Venkataramani. Dtn routing as a resource allocation problem. In *SIGCOMM 2007*.

[4] E. Cohen and S. Shenker. Replication strategies in unstructured peer-to-peer networks. In *SIGCOMM 2002*.

[5] W. Gao, Q. Li, B. Zhao, and G. Cao. Social-aware multicast in disruption-tolerant networks. 2012.

[6] R. Groenevelt, P. Nain, and G. Koole. The message delay in mobile ad hoc networks. *Performance Evaluation*, 62(1):210–228, 2005.

[7] Z. Guo, B. Wang, and J.H. Cui. Prediction assisted single-copy routing in underwater delay tolerant networks. In *GLOBECOM 2010*.

[8] S. Ioannidis, A. Chaintreau, and L. Massoulié. Optimal and scalable distribution of content updates over a mobile social network. In *INFOCOM 2009, IEEE*, pages 1422–1430. IEEE, 2009.

[9] S. Ioannidis, L. Massoulie, and A. Chaintreau. Distributed caching over heterogeneous mobile networks. *Queueing Systems*, pages 1–31, 2010.

[10] T. Karagiannis, J.Y. Le Boudec, and M. Vojnovic. Power law and exponential decay of intercontact times between mobile devices. *Mobile Computing, IEEE Transactions on*, 9(10):1377–1390, 2010.

[11] A. Krifa, C. Barakat, and T. Spyropoulos. Message drop and scheduling in dtns: Theory and practice. *Mobile Computing, IEEE Transactions on*, 11(9):1470–1483, 2012.

[12] Q. Li, S. Zhu, and G. Cao. Routing in socially selfish delay tolerant networks. In *INFOCOM 2010*.

[13] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. *In SIGMOBILE 2003*.

[14] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.

[15] J. Reich and A. Chaintreau. The age of impatience: optimal replication schemes for opportunistic networks. In *CoNEXT 2009*.

[16] P.J. Reny. Non cooperative games: Equilibrium existence. *The New Palgrave Dictionary of Economics, Second Edition*, 2005.

[17] M. Sathiamoorthy, A.G. Dimakis, B. Krishnamachari, and F. Bai. Distributed storage codes reduce latency in vehicular networks. In *INFOCOM-Mini 2012*.

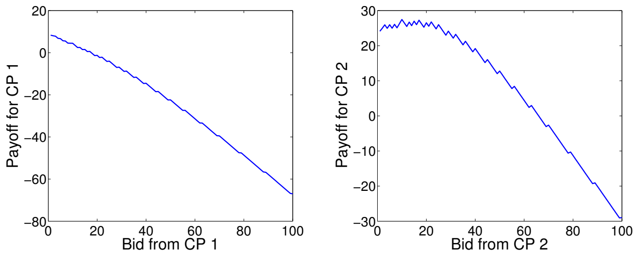
[18] M. Sathiamoorthy, W. Gao, B. Krishnamachari, and G. Cao. Minimum latency data diffusion in intermittently connected mobile networks. In *VTC Spring 2012*.

[19] U. Shevade, H.H. Song, L. Qiu, and Y. Zhang. Incentive-aware routing in dtns. In *ICNP 2008*.

[20] T. Small and Z.J. Haas. Resource and performance tradeoffs in delay-tolerant wireless networks. In *SIGCOMM 2005*.

[21] T. Spyropoulos, K. Psounis, and C.S. Raghavendra. Performance analysis of mobility-assisted routing. In *Mobicom 2006*.

[22] T. Spyropoulos, K. Psounis, and C.S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *SIGCOMM 2005*.




(a) $c_2$ fixed at 50 (b) $c_1$ fixed at 50

Fig. 9: Demonstrating a case where quasiconcavity does not hold. Here $c_0 = 1$ and $w = 10$.


[23] T. Spyropoulos, K. Psounis, and C.S. Raghavendra. Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility. In *PerCom Workshops '07*, pages 79–85. IEEE, 2007.

[24] Y. Wang, B. Krishnamachari, and T.W. Valente. Findings from an empirical study of fine-grained human social contacts. In *WONS 2009*.

[25] L. Wei, Z. Cao, and H. Zhu. Mobigame: A user-centric reputation based incentive protocol for delay/disruption tolerant networks. In *GLOBECOM 2011*.

[26] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance modeling of epidemic routing. *Computer Networks*, 51(10):2867–2891, 2007.


## Appendix

While previously we were able to show the existence of multiple Nash equilibria for parameters that were of interest to us, here we investigate the existence of Nash equilibria further in a more general setting. Specifically we determine regions of $w$ which guarantee the existence of Nash Equilibria for a few $c_0$ values. As before we consider the case of three players - one agent and two content providers for ease of understanding. Therefore $m = 2$ here.

We use the result from Theorem 2.2 in [16] here to discuss sufficient conditions for the existence of Nash equilibrium. We first note that the set of pure strategies is compact and convex. Consider player $i \in [m]$ who bids $c_i$. We fix the strategies of other players to $c_{-i}$ and would like to determine the conditions under which the payoff of player $i$, $P_i$ is quasiconcave with respect to $c_i$ (from the theorem, quasiconcavity guarantees the existence of Nash Equilibria).

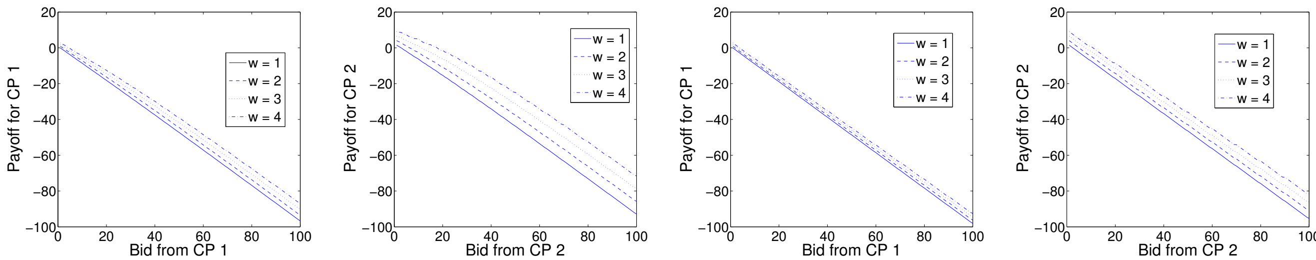
$P_i$ (a function of $c_i$) is quasiconcave if either (i) it is nondecreasing, (ii) it is nonincreasing, or (iii) there exists a $c_i^*$ such that $P_i$ is nondecreasing for $c_i < c_i^*$ and nonincreasing for $c_i > c_i^*$. While this can be checked by taking a derivative of $P_i$ with respect to $c_i$, since the expression for $P_i$ is not suitable for differentiation, we resort to numerically study the quasiconcavity property. Note that this can be verified rather easily numerically.

In Fig 7, we show a few plots of $P_1$ and $P_2$ for $c_0 = 1$ and $c_0 = 10$ for various $w$, where quasiconcavity is satisfied for model M1 (when the deadline is $T = 250$). Fig 8 shows a similar set of plots for model M2.

For the model M1, for the choices of $c_0 = 1$ and $c_0 = 10$, we see that $P_1, P_2$ are quasiconcave (in this case non increasing) when $w \leq 4$. For higher values of $w$, quasiconcavity does not hold (see Fig 9. Similarly, for the model M2, for the choices of $c_0 = 1$ and $c_0 = 10$, quasiconcavity is observed for $w \leq 0.008$. Since quasiconcavity is only a sufficient condition, higher values of $w$ do not necessarily preclude the existence of Nash equilibria.

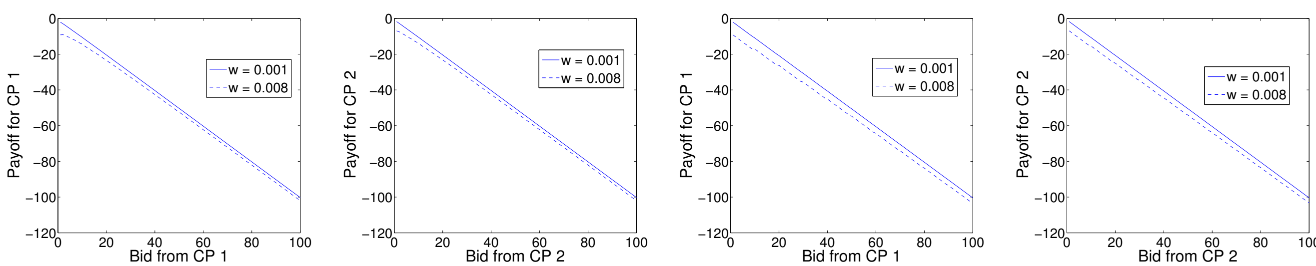
Fig 9 shows a case where the quasiconcavity property does not hold for both $P_1$ and $P_2$. Note that even though the curve in Fig 9a looks like it is decreasing, there are cases where the payoff decreases and then increases.

Investigation of the quasiconcavity of the payoffs analytically is left as a future work.




(a) $c_2$ fixed at 50, $c_0 = 1$ (b) $c_1$ fixed at 50, $c_0 = 1$ (c) $c_2$ fixed at 50, $c_0 = 10$ (d) $c_1$ fixed at 50, $c_0 = 10$

Fig. 7: A set of cases for model M1 (deadline $T = 250$ encounters) when quasiconcavity holds for both $P_1$ and $P_2$.




(a) $c_2$ fixed at 50, $c_0 = 1$ (b) $c_1$ fixed at 50, $c_0 = 1$ (c) $c_2$ fixed at 50, $c_0 = 10$ (d) $c_1$ fixed at 50, $c_0 = 10$

Fig. 8: A set of cases for model M2 when quasiconcavity holds for both $P_1$ and $P_2$.