Minimum Latency Data Diffusion in Intermittently Connected Mobile Networks

Maheswaran Sathiamoorthy*, Wei Gao†, Bhaskar Krishnamachari*, Guohong Cao†
*Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, CA
†Department of Computer Science and Engineering, Pennsylvania State University, State College, PA
Email: {msathiam, bkrishna}@usc.edu,{weigao, gcao}@cse.psu.edu

***Abstract*—We consider the problem of diffusing cached content in an intermittently connected mobile network, starting from a given initial configuration to a desirable goal state where all nodes interested in particular contents have a copy of their desired contents. The goal is to minimize the time taken for the diffusion process to terminate at a goal state. Due to bandwidth and storage constraints, whenever two nodes encounter each other, they must decide which content if any to transfer to each other. While most prior work on this topic has focused on practically realizable heuristics for this problem, we take a more formal approach. Our main contribution is to show that, assuming global state information is available, this problem can be formulated as a stochastic shortest path problem, which is a kind of Markov decision process (MDP). Using this formulation, we numerically explore some small-scale examples for which we are able to obtain the optimal solution. The results show that the optimal diffusion strategy is very much a function of the underlying encounter graph.**

I. Introduction

With a rising number of mobile devices and surging interest in vehicular networks, there is a compelling use case for intermittently connected mobile network (ICMN) based data dissemination. For it to be effective, nodes must participate in routing data, sometimes even if they are not interested in the data. In such settings, routing may often involve storing data until it is handed over to some other node. Thus questions arise as to how to route data when there are storage and bandwidth constraints and nodes are not interested in storing all the data. Clearly, if there were no storage constraints, nodes can store all the data, including those that they are not interested in and pass the data over to other nodes later. But when there is a storage constraint, when two nodes meet, what should a node in question download and what content should it possibly evict? Clearly, selfish strategies involving nodes trying to store only the data they are interested in either does not work or will not be effective. In the past, various heuristics have been suggested that use social contact information. In our work, we formalize the problem and express it as a stochastic shortest path problem (SSP) [1]. Although this formulation is computationally intractable, we are able to solve it exactly for small settings which can give useful insights.

The paper is organized as follows. In the following section, we discuss relevant literature on data dissemination in ICMNs, and in Section III, we describe a few assumptions and the model used for the ICMN. While Section IV formulates the data diffusion problem as an SSP, Section V shows the results of a few simulations along with a few insights. The work is concluded in VI.

This research was sponsored by the U.S. Army Research Laboratory under the Network Science Collaborative Technology Alliance, Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

II. Related Work

The fundamental goal of this paper is to model the data diffusion process in an ICMN into a mathematical framework. In particular, we model the diffusion of data as a stochastic shortest path problem so that it can be solved for minimum delay. In contrast to this paper, most of the previous works focus on heuristics that are realizable but may not be optimal. In fact, there is a rich literature on routing for ICMN, which can roughly be classified into two classes. Most of the earlier works focus on pushing data to select destinations agnostic of the underlying social network, such as Epidemic routing [2], Spray and Wait [3], Spray and Focus [4] etc. Recently however, there have been works on using social aspects of users behaviour to ease the propagation of the messages.

While the epidemic routing is flooding tailored for ICMNs, Spray and Wait and Spray and Focus achieve better performance by systematically restricting the number of packets in the network. The scope of these works are limited in a number of ways - the destination is usually a single node; it is not clear how to route when data is needed by multiple nodes. Also storage and bandwidth constraints are not considered. For example, if a node receives one of the messages from the spray stage when its buffer is full, should it accept it or which other message should it remove from its buffer? This is clearly a non-trivial problem.

The second class of works, directly relevant to ours, are the socio-aware diffusion schemes. Zhang *et al.* [5] propose four heuristics based on both the social relationships between contacts and the data similarities. Of these, they identify that the one scheme that diffuses similar content between friends and dissimilar content between strangers works better than other schemes. Their work relies on the homophily phenomenon [6] which states that similar people tend to become friends. They further assert that by having similar interests, friends will tend to meet often and encounters between strangers are less likely. While it is a tractable assumption, it may not be exactly true in practice. In their subsequent work [7], they use a social centrality metric based simultaneuosly on contact patterns and interests and in [8], social-aware multicasting is considered.

In ContentPlace [9], each node assigns a utility value to each data object and when there is an encounter, data items are fetched that maximize the local utility. These utilities are calculated based on how much the objects are desired by communities that the node visits often. Similar work predating ContentPlace are PodNet [10] and Haggle [11]. In PodNet, the authors built a podcasting system for wireless ad-hoc networks over mobile phones. When a node interested in a particular data item meets another node that has this data, the former downloads from the latter. In Haggle, communities and their hubs are identified and an overlay network is built connecting the hubs which facilitates the data dissemination.

The unique new contribution here is that ours is the first work to formulate the data dissemination problem in ICMNs into a mathematical framework and solve it.

III. Models and Assumptions

A. Setup

Consider a system consisting of N users or nodes, K data objects $d_1, d_2, \ldots, d_K$ each of unit size, and each user i ($i = 1, 2, \ldots, N$) has capacity $\mathcal{C}$ and is interested in at most $\mathcal{C}$ data objects. We will consider a node happy if it has all the data objects it is interested in. Let E be the matrix indicating which node has what, also called *data-incidence matrix*. Element (i, k) indicates whether node i has data object k or not, depending on if the value is 1 or 0. We do not consider fractional amount of data and we assume that whenever a download is attempted, a full file can be downloaded. When all the nodes have what they want, we can denote the data-incidence matrix as E_t, which can also be called as *interest matrix* since it represents what the nodes are interested in.

B. Contact Process

In this work, we use the following simplified model for the contact processes: node encounters are separated by *i.i.d.* time slots and during each encounter, two distinct nodes are chosen randomly with some predefined probability to meet with each other. Further, during each encounter, let us also assume that only one of the two nodes will be able to modify its cache (download and/or remove or do nothing). Thus the encounters are ordered and there are $N(N-1)$ possible encounters. We use the ordered pair notation (i, j) to indicate that the encounter is between i and j and that i downloads from j. Each encounter (i, j) can be assigned a non-negative probability p_{ij} (so that $\sum_{i \neq j} p_{ij} = 1$). If the inter-encounter durations are represented by a random variable T with expectation $\mathbb{E}\{T\}$, then the total duration for m encounters will be proportional to $\mathbb{E}\{T\}$ in expectation. Thus, it is enough to calculate the number of encounters required to get all the nodes what they desire.

The goal of this work is to find out strategies that enable nodes to reach their above states of content even if they begin with arbitrary content distribution (also integral). Thus, given any E, we want the system to move to E_t.

IV. Stochastic Shortest Path formulation

In this section we describe how the data dissemination problem can be modeled as a stochastic shortest path problem, which is a special class of markov decision process (MDP) [1]. An SSP problem ρ is represented by a 5-tuple $\{\mathcal{S}, t, A, P.(\cdot,\cdot), C.(\cdot,\cdot)\}$ where

1) $\mathcal{S}$ is the state space, consisting of a finite number of states.
2) $t \in \mathcal{S}$ is a terminating state (in general there can be many).
3) A is a finite set of actions.
4) P_a is a transition probabiliity matrix governing the state transitions once action a is taken.
5) $C_a(s, s')$ is the non-negative cost associated with transitioning from state s to state s' having taken action a.

The goal is then, given a starting state, to find out policies that minimize the total non-discounted cost from any state to a terminating state. Note that the total cost is non-discounted, unlike many other MDP formulations.

A. State Space

During each encounter, which content is to be downloaded and/or removed, if any, depends not just on the encountering nodes but also on who else needs the same content in the future. So it is clear that the data-incidence matrix must be used as a descriptor of the state, in addition to who downloads from whom. Thus let $s = \{E, (i, j)\}$ denote a non-terminating state of the system at any time, where E is the data-incidence matrix and the ordered pair (i, j) denotes that users i and j meet each other and that i downloads from j at most one data item. In addition we also have a terminating state, which is described next. Since the goal is to get all the nodes their desired content, the data-incidence matrix upon termination must be E_t. This we consider as the single terminating state $t = \{E_t\}$ and set $P_a(t, t) = 1, \forall a \in A$. For the terminating state, the encounters are irrelevant and hence are removed from the state description. If $\mathcal{E}$ is the set of all possible $N \times K$ data-incidence (binary) matrices, then

$$
\begin{aligned}
\mathcal{S} = &\{\{E, (i, j)\} : E \in \mathcal{E} \backslash E_t, i, j \in \{1, 2, \ldots, N\}, i \neq j\} \\
&\cup \{\{E_t\}\}
\end{aligned}
$$

Note that some of the non-terminating states are *illegal* if data items are lost completely from system or if some node stores more than its capacity. For example, clearly when the incidence matrix of a state is an $N \times K$ matrix with all zeros or all ones, then the state must be illegal.

B. Actions

If the current state of the SSP is a non-terminating state, say $s = \{E, (i, j)\}$, the action specifies which data item i must remove if any from its storage, and the data item to

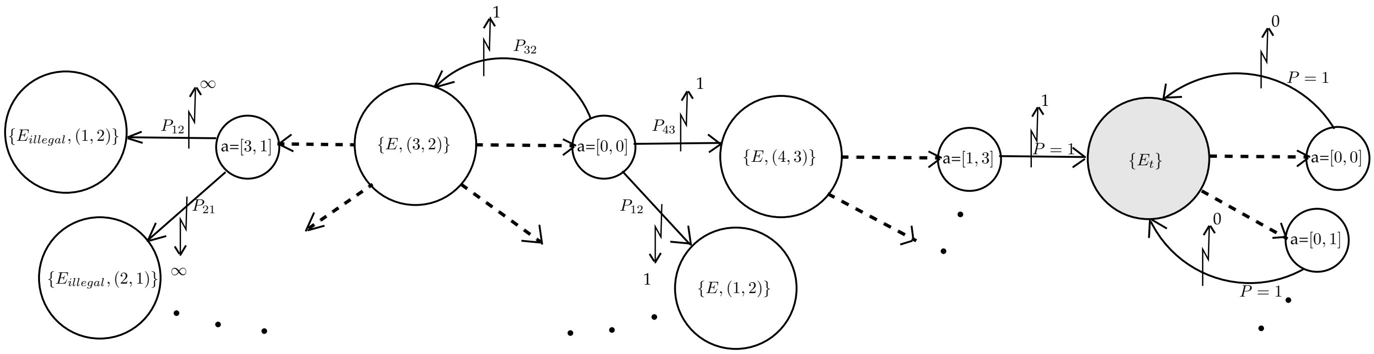


Fig. 1. Example State Dynamics. The big circles are states and the smaller ones are actions. Consider the state $\{E,(3,2)\}$ in which node 3 downloads from node 2. E is the data-incidence matrix as defined in Eq 2. If action $[0,0]$ is performed, then there is no change in the data-incidence matrix and only the encounters change in the next state. But if the action was $[3,1]$, object 3 would be removed to accommodate object 1 because the capacity is only two, thus 3 would be lost completely from the system, leading to illegal states. Now consider state $\{E,(4,3)\}$. When node 4 downloads from 3, if it replaces object 1 with object 3, all nodes will end up with what they desire, so that the system can be terminated. Note that the terminating state is equivalent to the interest matrix.

download from j, if needed. This is represented by $a = [d_{\text{remove}}, d_{\text{download}}]$, where $d_{\text{remove}}, d_{\text{download}}$ take values between 0 and K. If d_{remove} is 0, the node doesn't remove any data item, else if it is k, it will remove data item d_k. Similarly if d_{download} is 0, the node will not download anything, otherwise if it is k, it will download data item k. Once an action is applied, the system moves possibly to a new state and so on. When the terminating state $\{E_t\}$ is reached, the SSP is stopped.

C. Costs

Each transition has a cost associated with it. Since the objective is to count the number of steps it takes for the SSP to reach the terminating state (and minimize it), the cost of all the transitions are set to 1, with a few exceptions as below. Transitioning to an illegal state is prohibited by setting the cost to ∞ and the cost from the terminating state to any other state is set to 0. For all actions, we have,

$$C_a(s,s') = \begin{cases} 0 & \text{if } s \text{ is a terminating state} \\ \infty & \text{if } s' \text{ is an illegal state} \\ 1 & \text{else} \end{cases}$$

D. The Markovian Dynamics

Let us define the probabilities associated with transitioning from one state to another, given a particular action is performed. Let the current state be $s = \{E,(i,j)\}$. If s is a terminating state, set $P_a(s,s) = 1, \forall a \in A$ so that the SSP effectively stops at the terminating state. Otherwise if s is not a terminating state, the action will transform E to E'. Let $E' \neq E_t$. Now a new encountering pair will be chosen from the distribution defined above. Thus if the probability of choosing pair (i',j') is $p_{i'j'}$, then $P_a(s,s') = p_{i'j'}$ for $s' = \{E',(i',j')\}$. If $E' = E_t$, then set $P_a(s,s') = 1$ (this is because the terminating state is agnostic of the encounters).

E. Four Node Example

Let the number of nodes $N = 4$, the number of data objects $K = 4$ and capacity per node $\mathcal{C} = 2$. Let nodes $1,2$ desire objects d_1 and d_2, 3 desires d_3 and d_4, and 4 desires d_2 and d_3. So the interest matrix is

$$E_t = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \tag{1}$$

which also defines the terminating state $\{E_t\}$. Also denote any data-incidence matrix

$$E = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}. \tag{2}$$

Let the data-incidence matrix corresponding to an illegal state be

$$E_{\text{illegal}} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix} \tag{3}$$

since data object d_3 has been completely lost. Let each ordered encounter (i,j) occur with probability p_{ij}. Some of the states from the SSP, along with a few actions, transition probabilities and transition costs are shown in Fig 1.

F. Dynamic Program Formulation

An SSP ρ can be expressed as a stochastic dynamic program [1], which we attempt to describe in this section. Let $J(s)$ be the minimum total non-discounted expected cost to go from s to the terminating state. Applying an action a on state s can take the system to a state s' with probability $P_a(s,s')$. The cost to go from s to s' is $C_a(s,s')$ and the minimum total expected cost to go from s' to the terminating state is $J(s')$. Thus, action a has a total expected cost of $\sum_{s'} P_a(s,s')\left[C_a(s,s') + J(s')\right]$. Thus,

$$J(s) = \min_a \sum_{s'} P_a(s,s')\left[C_a(s,s') + J(s')\right]. \tag{4}$$

And the optimal action for s, denoted by $\Pi(s)$ is therefore

$$\Pi(s) = \arg\min_a \sum_{s'} P_a(s,s')\left[C_a(s,s') + J(s')\right]. \tag{5}$$

In particular, $J(s) = 0$ and ∞ for the terminating and the illegal states respectively. First we note that there can be network mobility patterns and data configurations such that the values may not converge. We do not consider such cases. Second, this dynamic program can be solved using value iteration. In many cases the values may not converge in finite steps, so we choose an error tolerance and terminate after dropping below the tolerance.

V. Simulations and Results

Since the number of states grows exponentially with the number of users, we limit our simulations to 4 users. One of the primary objectives of the simulations is to understand how the optimal strategy works and to derive insights.

A. Simulation setup

We consider two identical networks each consisting of four nodes, but differing only in the contact probabilities. Let $N = 4, K = 4$ and $\mathcal{C} = 2$. Let the interest-matrix, which corresponds to the terminating state in both the networks be the same as in Eq 1. We vary the contact probabilities to obtain two different network setups shown in Fig 2(a) and Fig 2(b). Essentially, there are two types of contacts, high probability contacts (the ones with thick lines) and low probability contacts (those with the thin lines). We fix the values of all high probability contacts to be the same to h and all the low probabilities have the same value of l. When there is no edge between nodes, it means that the two nodes do not meet. For both the setups, we use a high to low probability ratio of 9:1, and bidirectional arrows between two nodes i and j indicate that the probabilities of (i, j) and (j, i) are the same (either h or l each).

In order to understand the socio-aware schemes, we consider friendship relationships by using the homophily phenomenon. Thus nodes that share common interests are noted as friends (F) whereas nodes that have nothing in common are noted as strangers (S). If there is a partial overlap, we just denote them as Strangers/Friends (S/F).

For any ordered encounter (i, j), the actions can be categorized into the following events based on what i does:

- A_1 — i downloads content it needs.
- A_2 — i downloads content it doesn't need.
- A_3 — i doesn't download anything.
- B_1 — i downloads content it needs, given that j has what it needs.
- B_2 — i downloads content it doesn't need, given that j has what it needs.
- B_3 — i doesn't download anything, given j has what it needs.
- C_1 — i downloads content it needs, given that j has what i needs and i doesn't have it itself.
- C_2 — i downloads content it doesn't need, given that j has what i needs and i doesn't have it itself.
- C_3 — i doesn't download anything, given that j has what i needs and i doesn't have it itself.

We will find out the normalized frequency of these events for all possible ordered encounters (i, j). The frequencies are calculated by assigning equal weight for all states where the ordered encounter (i, j) occurs and counting for the occurrence of desired events. For example, suppose we want to calculate the frequency of event B_2 for encounter $(1, 2)$. First we count the number of states in which there is a $(1, 2)$ encounter and where node 2 has data d_1 or data d_2 (items node 1 needs) and then among these states we count the number of times when the optimal action makes node 1 download d_3 or d_4 (items node 1 does not need) instead of d_1 or d_2. All the frequencies for both the setups are tabulated in Table I and Table II and the results are explained in the following sections.

B. First setup

The setup is shown in Fig 2(a). In this setup, there is a direct correlation between friendship statuses and contact frequencies. So we let friends meet very often while making strangers meet less often. The S/F contacts are made to meet both often as well as rarely. The frequencies for the various events identified are shown in Table I. Note that there are some values in the table pertaining to contacts that never occur; for example, nodes 1 and 3 never meet each other, but nevertheless the SSP can still solve for all possible encounters.

First we note that the frequencies for node 1 when it meets nodes 2, 3 or 4 are roughly the same, irrespective of whether the node it meets is a friend or a stranger or both. The same holds for node 3. So we claim that what a node downloads depends a lot on who else the node is likely meet later rather than the friendship status.

In this regard, node 1 only meets node 2 most often (and occassionally node 4). This is also the reason why node 2 or node 4's column frequencies are not similar.

Let us compare the frequencies for nodes 1 and 4. When node 1 meets 2, it downloads content it needs (A_1) with frequency 0.4445 whereas when 4 meets 2, the corresponding frequency is a bit lower at 0.3389. This is because node 1 is mostly independent and doesn't have to buffer data for other nodes (since it only meets node 2 often with which it shares all the interests). Whereas node 4 has to buffer data for node 3 and so the frequency of downloading what it wants is lower. Also node 4 downloads content it doesn't need even when it doesn't have the content it needs (C_2) with frequency 0.1136. In contrast this frequency is 0 for node 1.

So it is clear that what a node downloads depends on what it wants and also on what its neighbors want. This also explains why the columns corresponding to $(2, 1)$ and $(2, 3)$ are markedly different. When node 2 meets node 3, it doesn't have to buffer data for others as much as it has to buffer uninterested data from node 1.

C. Second setup

The setup is shown in Fig 2(b) and the results shown in Table II. In this setup, any correlation between friendship status and contact frequency is removed by letting both friends and strangers meet often (in some cases). For example, nodes 2

Event*	Freq. when node 1 meets			Freq. when node 2 meets			Freq. when node 3 meets			Freq. when node 4 meets		
	node 2 (F)	node 3 (S)	node 4 (S/F)	node 1 (F)	node 3 (S)	node 4 (S/F)	node 1 (S)	node 2 (S)	node 4 (S/F)	node 1 (S/F)	node 2 (S/F)	node 3 (S/F)
A_1	0.44	0.47	0.46	0.33	0.44	0.42	0.46	0.46	0.45	0.36	0.34	0.35
A_2	0.14	0.08	0.07	0.29	0.08	0.13	0.16	0.11	0.14	0.23	0.23	0.17
A_3	0.42	0.45	0.47	0.38	0.48	0.45	0.38	0.44	0.41	0.41	0.43	0.48
B_1	0.64	0.67	0.67	0.47	0.63	0.61	0.66	0.66	0.64	0.51	0.49	0.51
B_2	0.05	0.03	0.01	0.22	0.02	0.04	0.06	0.03	0.04	0.16	0.16	0.11
B_3	0.31	0.30	0.32	0.31	0.35	0.35	0.28	0.31	0.32	0.33	0.35	0.38
C_1	0.87	0.92	0.91	0.64	0.86	0.82	0.90	0.89	0.87	0.70	0.67	0.69
C_2	0.00	0.00	0.00	0.17	0.00	0.01	0.01	0.00	0.00	0.11	0.11	0.08
C_3	0.13	0.08	0.09	0.19	0.14	0.17	0.09	0.11	0.13	0.19	0.22	0.23

*Note that the encounters are ordered; so when node 1 meets node 2, node 2 will is restricted to be a source.

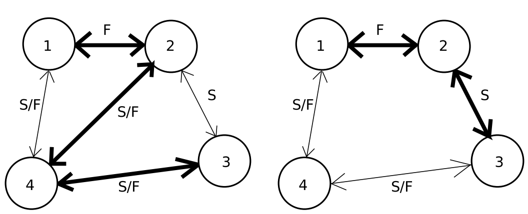
TABLE I
RESULTS OF SETUP 1

Event*	Freq. when node 1 meets			Freq. when node 2 meets			Freq. when node 3 meets			Freq. when node 4 meets		
	node 2 (F)	node 3 (S)	node 4 (S/F)	node 1 (F)	node 3 (S)	node 4 (S/F)	node 1 (S)	node 2 (S)	node 4 (S/F)	node 1 (S/F)	node 2 (S/F)	node 3 (S/F)
A_1	0.34	0.36	0.36	0.22	0.34	0.34	0.37	0.33	0.36	0.46	0.46	0.46
A_2	0.19	0.19	0.26	0.32	0.17	0.31	0.21	0.20	0.28	0.15	0.15	0.13
A_3	0.47	0.45	0.38	0.46	0.49	0.35	0.42	0.47	0.36	0.39	0.39	0.41
B_1	0.50	0.51	0.52	0.32	0.49	0.49	0.53	0.47	0.52	0.66	0.66	0.66
B_2	0.11	0.12	0.17	0.28	0.09	0.22	0.12	0.13	0.19	0.05	0.05	0.04
B_3	0.39	0.37	0.31	0.40	0.42	0.29	0.35	0.40	0.29	0.29	0.29	0.30
C_1	0.68	0.70	0.71	0.43	0.67	0.66	0.72	0.64	0.71	0.90	0.90	0.90
C_2	0.07	0.07	0.12	0.21	0.05	0.16	0.06	0.09	0.13	0.00	0.00	0.00
C_3	0.26	0.23	0.17	0.36	0.28	0.18	0.22	0.27	0.16	0.10	0.10	0.10

*The encounters are ordered as before.

TABLE II
RESULTS OF SETUP 2

and 3 which are strangers have high contact frequency. When node 2 meets node 1, the frequency of event C_2 is 0.2087, which is quite high; this is because node 2 has to download data that node 3 is interested in. If not for node 2, node 3 will have a slow route through node 4 and so the SSP solution encourages node 2 to buffer uninterested data for node 3. Here again we see that the contact probabilities matter but not the friendship relationships.



(a) First setup (b) Second setup

Fig. 2. Two contact graphs on the same set of nodes, but each having different contact probabilities. The thick lines indicate high probability of contact and thin lines indicate low probabilities. Labels S,F and S/F refer to Strangers, Friends or Strangers/Friends. More details in Section III-A.

VI. Conclusion

In this work, we have modeled the data diffusion process in ICMNs as a novel Stochastic Shorest Path problem. This is important from a theoretical point of view to understand the problem and to gain insights on the solution. Further, it may be possible to obtain better heuristics based on the solution, which we will continue to investigate in the future. Another direction is to possibly find out methods to solve the SSP faster.

References

[1] D. Bertsekas, "Dynamic programming and optimal control," *Athena Scientific Belmont, MA*, 1995.

[2] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," CS-2000-06, Duke University, Tech. Rep., 2000.

[3] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proc. of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. ACM, 2005, pp. 252–259.

[4] ——, "Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility," in *PerCom Workshops '07*. IEEE, 2007, pp. 79–85.

[5] Y. Zhang, W. Gao, G. Cao, T. La Porta, B. Krishnamachari, and A. Iyengar, "Social-aware data diffusion in delay tolerant manets," *Handbook of Optimization in Complex Networks Communication and Social Networks*, 2011.

[6] M. McPherson, L. Smith-Lovin, and J. Cook, "Birds of a feather: Homophily in social networks," *Annual review of sociology*, pp. 415–444, 2001.

[7] W. Gao and G. Cao, "User-centric data dissemination in disruption tolerant networks," in *Proc. of Infocom*, 2011.

[8] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: a social network perspective," in *Proc. of MobiHoc*. ACM, 2009, pp. 299–308.

[9] C. Boldrini, M. Conti, and A. Passarella, "Contentplace: social-aware data dissemination in opportunistic networks," in *Proc. of MSWiM*. ACM, 2008, pp. 203–210.

[10] V. Lenders, G. Karlsson, and M. May, "Wireless ad hoc podcasting," in *Proc. of SECON*, june 2007, pp. 273–283.

[11] E. Yoneki, P. Hui, S. Chan, and J. Crowcroft, "A socio-aware overlay for publish/subscribe communication in delay tolerant networks," in *Proc. of MSWiM*. ACM, 2007, pp. 225–234.