Design and Analysis of Self-Organized Data Aggregation Using Evolutionary Game Theory in Delay Tolerant Networks

K. Habibul Kabir, Masahiro Sasabe, and Tetsuya Takine
Department of Information and Communications Technology, Graduate School of Engineering,
Osaka University, 2-1 Yamadaoka, Suita-shi, Osaka 565-0871, Japan
habib@post.comm.eng.osaka-u.ac.jp, {sasabe, takine}@comm.eng.osaka-u.ac.jp

Abstract

In delay tolerant networks (DTNs), custody transfer mechanism provides reliable end-to-end data transfer in which special nodes (custodians) transfer data with custody in a hop-by-hop manner. As a result, storage congestion occurs when data with custody increases and/or the network is partitioned into multiple sub-networks for a long time. The storage congestion can be alleviated with the help of message ferries. In such a scenario, data should be aggregated to some custodians so that message ferries can effectively collect them. In this paper, we propose a scheme to aggregate data into selected custodians, called aggregators, in a fully distributed and autonomous manner by using evolutionary game theoretical approach where we can also control the number of aggregators to a desired value.

1 Introduction

With the development of networking technologies, many researchers and developers have tried to achieve data communications in challenged networks, called delay tolerant networks (DTNs) [1,2]. In DTNs, a *store-carry-forward* [1] message delivery mechanism is used. A source node combines multiple data into a bundle and transmits it to the destination node in a hop-by-hop manner. However, instantaneous acknowledgment cannot be obtained due to lack of permanent end-to-end connectivity. *Custody transfer* [3] mechanism ensures reliable data transfer among nodes in DTNs. Custody transfer offers that a bundle with custody must be perfectly delivered from a source to the destination by delegating the responsibility of reliable transfer with the bundle in a hop-by-hop manner. Intermediate nodes keeping bundles with custody are called *custodians*. Note here that to be a custodian, a node must reserve a sufficient amount of storage and energy to receive bundles with custody and hold them until successful delivery or the expiration of the bundle's delivery time. Congestion occurs when storage resources become scarce due to too many bundles with custody. An increase of bundles with custody and long-term network partitioning accelerate the storage congestion.

To solve the storage congestion problem, some special mobile nodes can be introduced to proactively travel the network and gather bundles from custodians before the congestion occurs. These special mobile nodes are referred to as *Message Ferries* [12, 13]. If the network is divided into several isolated networks (clusters), message ferries move around the deployment area and deliver bundles among the clusters. There are two message ferry schemes [13]: node-initiated message ferry and ferry-initiated message ferry. In the ferry-initiated message ferry scheme, message ferries take proactive movement to meet the custodians. After receiving the service request from a custodian, the message ferry moves to the custodian and collects bundles. It can also supply energy to the custodian if required. However, if the requests from storage congested nodes increase, sometimes it is hard for the message ferry to visit all of them in a certain period of time. Although increasing the number of message ferries can cope with this problem, it also introduces additional installation costs on the system.

In DTNs, any custodian cannot predict how long it should keep bundles with custody. Note that each node in DTNs is basically powered by a battery and it has to be always awake when holding the bundles. Since each custodian also generates its own bundles with custody, it is selfish and rejects requests for custody transfer from other nodes to save its storage as well as its energy. This means that the custody transfer mechanism fails without taking the selfishness of custodians into account.

In summary, we face two challenges: a) The number of nodes that can be visited by the message ferry in a given period of time is limited, b) nodes are potentially selfish and are not willing to store others' bundles. To tackle these challenges, we propose a system that can a) gather all bundles in a partitioned network to some selected nodes in the network so that message ferries can collect them effectively and b)

978-1-4244-4439-7/09/$25.00 ©2009 IEEE

take the nodes' selfishness into account. The system should also be self-organized and decentralized because each node can only know its surrounding environment in DTNs.

To accomplish such a system, evolutionary game theoretic approach is one of the most appropriate mechanisms because of its rational strategic decision making characteristics. Evolutionary game theory explores the dynamics of a population of players under the influence of natural selection using a mathematical model called replicator dynamics [11]. It assumes that the more the fitness (payoff) is acquired, the larger the population of the corresponding species is.

Replicator dynamics on graphs [5–9] is an extension of the original replicator dynamics to a finite population where the members of the population represent the vertices of a regular graph. It describes how the expected frequency of each strategy changes over times within the graphs.

With the help of this scheme, we can select some special custodians referred to as *aggregators*, which are cooperative in nature and willingly hold bundles with custody of other nodes.

As the title indicates the design and analysis part of self-organized data aggregation, in this paper we mainly focus on the overall of the system design and the mathematical analysis of the applicability of evolutionary game theory in the challenged networks of DTNs.

2 Proposed Scheme

2.1 Overview

In this paper, we aim to achieve a system that periodically collects information from multiple isolated networks called clusters, e.g., several sensing areas in sensor networks, many evacuation sites in disaster areas, etc. We can model these scenarios as follows. The system consists of one or more sink nodes and lots of clusters. Each node can directly communicate only with other nodes in the transmission range. To collect bundles from the clusters to the sink node, we apply the ferry-initiated message ferry scheme [13], where the message ferry departs from the sink node, visits each cluster to gather bundles, and then brings them back to the sink node as shown in Fig. 1(a). The duration of this cycle should be as short as possible so that the sink node grasps the current conditions of all the clusters.

To shorten the period for collecting bundles in a cluster, we propose a scheme to aggregate bundles in each cluster to some nodes referred to as aggregators. In each cluster, the aggregators are autonomously selected from nodes in the cluster, called cluster members, by local interactions among them. Each non-aggregator (sender) sends its bundles to the aggregators so that the message ferry will be required to visit only the aggregators as illustrated in Fig. 1(b).

We can summarize the above scenario in each cluster as the repetition of the following three phases:

1. *Aggregator selecting phase* - Each node selects to be an aggregator or a sender based on local interactions with the neighboring nodes.
2. *Bundle aggregating phase* - When each sender generates its own bundles, it transmits them to one of the aggregators.
3. *Bundle collecting phase* - Each aggregator transmits its service request to the message ferry and sends all bundles to the ferry.

We define a *round* as the unit of this repetition as shown in Fig. 2(a). During each round, each node performs these three phases. We assume that all nodes synchronize each other and know the length of the round which is predetermined by the sink node. The length of the round can also be updated through the communication between the ferry and nodes if needed.

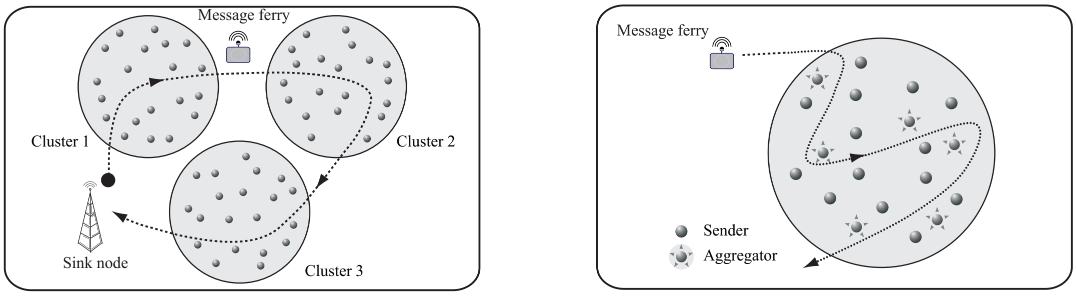
This scenario not only shortens the duration of the round but also gives all nodes benefits in terms of long battery life. There are two ways to keep their batteries in high levels: 1) obtaining the battery supply from the message ferry at the end of the round, and 2) reducing the battery consumption by sleeping as long as possible in the round. The former (latter) case can be regarded as being an aggregator (a sender). Aggregators should be awake all the time in the round to receive bundles from senders as shown in Fig. 2(b). As a result, they consume much energy than senders but can also obtain the battery supply from the message ferry. Note that in the ferry-initiated message ferry scheme [13], each aggregator transmits its service request to the ferry by long range wireless radio, which also consumes much energy.

Taking account of these characteristics, we expect that the system works well under the conditions: 1) There exist a small number of aggregators and many senders, and 2) the role of a node should change per round as shown in Fig. 2(c).

These challenges can be divided into two problems: 1) How can aggregators be selected autonomously under situations where all nodes are potentially selfish?, 2) Can we control the number of aggregators? To cope with these problems, we apply evolutionary game theoretic approach. Evolutionary game theory explores the dynamics of the system from the local interactions among the nodes.

2.2 Selection of the Aggregators

Each node is under mutual dependency with other cluster members. If there are already one or more aggregators in the same cluster, it is better for the node to be a sender. Otherwise, it should become an aggregator. Since it is difficult to achieve a centralized control in DTNs due to lack of persistent connectivity among arbitrary nodes, the selection of



(a) Message ferry visits each cluster and delivers collected bundles to the sink node.

(b) Message ferry visits aggregators in a cluster.

Figure 1. Proposed scenario.

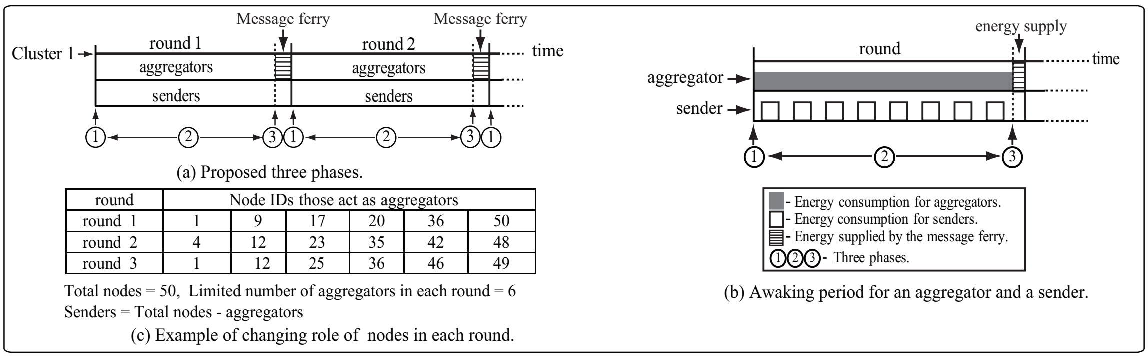


Figure 2. Example of proposed three phases, awaking period and role of each node.

aggregators should be realized in a decentralized way. We assume that each node communicates only with its neighbors and determines to be an aggregator or a sender based on its own benefit depending on the surrounding conditions.

We assume that each node loses energy proportional to the length of time it keeps awake. As mentioned above and illustrated in Fig. 2(b), aggregators are always awake during a round while senders only wake up when generating and transmitting their bundles. Let c and s denote the amount of energy consumption for aggregators and senders, respectively, per round. Obviously, $c \geq s$. s approaches c with the rate of generating bundles. The energy supplied by the message ferry to each aggregator is represented by b. Intuitively, the larger b is, the more the aggregators increases. $b \geq c$ should also be satisfied to suppress the number of senders. In summary, the benefit b is only a parameter for the sink node to control the number of aggregators in each cluster and should be carefully tuned by the sink node.

We first model the bargain among nodes as a game between two neighboring nodes in evolutionary game theory. There are two roles (strategies) for each node: aggregator (aggregate) and sender (send). There are four possible combinations of the strategies of the two nodes as in Table 1.

The resulting payoffs for each case can be modeled by taking the energy supply and energy consumption into account. If both nodes select to be aggregators, they lose the largest energy c without any energy supply from the message ferry, because they are not be able to collect a sufficient number of bundles to request the ferry to visit. An aggregator paired with a sender obtains the largest energy $b-c$; it loses c but obtains b from the message ferry. In this case, the corresponding sender loses the smallest energy s. In the last case, both nodes select to be senders and consume s (case 1). Here, they lose c in the worst case where the sender has to keep awake all the time in a round due to continuous retransmission (case 2).

We can abstract Table 1 into Table 2 where $T > S =$

Table 1. Payoff matrix.

node 1 \ node 2	send	aggregate
send	$-s\ (-c), -s\ (-c)$	$-s, b-c$
aggregate	$b-c, -s$	$-c, -c$

Table 2. Abstracted Payoff matrix.

node 1 \ node 2	send	aggregate
send	R, R	S, T
aggregate	T, S	P, P

$R > P$ (case 1) or $T > S > R = P$ (case 2). In both cases, every node not only has a temptation to be an aggregator ($T > R$) but also a fear to be an aggregator ($S > P$). The larger b is, the more the temptation is. This indicates that the sink node can control the number of aggregators (senders) by changing b. We show the detail in Section 3. The condition $T > R$ and $S > P$ also has another significant characteristic; taking a strategy different from the opponent is better than taking the same strategy as the opponent. As a result, both aggregating and sending strategies stably coexist [5]. Thus, with the help of the payoff-matrix and evolutionary game theory, when each node undertakes suitable strategies to optimize its own payoff, then the system converges to a fully stable situation where both senders and aggregators stably coexist.

One may think that the proposed scheme is similar to LEACH [4] which is a cluster-head selection scheme in wireless sensor networks. The aims of both schemes are almost the same but LEACH assumes that all nodes cooperatively behave each other. On the other hand, in the proposed scheme, the aggregators' selection totally depends on the nodes' mutual interactions by taking account of selfishness of each node. Thus, the proposed scheme is also applicable to the cluster-head selection in a more robust manner.

3 Analytical Results

In this section, we analytically derive the relationship between the ratio of aggregators and the parameters of the payoff matrix through replicator equation on graphs of evolutionary game theory [6–8]. Each node derives a payoff from the interactions with all of its connected neighbors. Then it compares the obtained payoff with a randomly chosen neighbor. If it overcomes the opponent, it keeps the current strategy. Otherwise, it imitates the strategy of the opponent. The basic concept of replicator dynamics is that the growth rate of nodes taking a specific strategy is proportional to the payoff acquired by the strategy. Thus the strategy that yields more payoff than the average payoff of the whole system increases. Replicator dynamics on graphs additionally takes account of the effect of the topological structure of the network which is suitable for the DTNs.

3.1 Replicator Equation on Graphs

First, we derive the replicator equation on graphs [8] for case 1 as described in Table 1. Let x denote the ratio of the number of aggregators to the total number of cluster members. Note that $1 - x$ represents the ratio of the number of senders. The expected payoff (fitness) f_1 and f_2 of aggregators and senders are given by

$$f_1 = (1-x)(b-c) - cx, \quad f_2 = -s, \tag{1}$$

Table 3. Modifier matrix.

node 1 \ node 2	send	aggregate
send	$0, 0$	$-m, m$
aggregate	$m, -m$	$0, 0$

respectively.

Let k denote the number of neighbors of each node, called degree [6]. Although we will present the analysis based on the k-regular graph as in Ref. [8], the analysis is also applicable to non-regular graphs, e.g., scale free networks [6, 8]. In such case, k denotes the average degree. The modified payoff matrix for evolutionary game theory on graphs is defined as the sum of the original payoff matrix and a modifier matrix [8]. Table 3 shows the modifier matrix. For case 1, m becomes [8]

$$m = \frac{3b - (k+6)(c-s)}{(k+3)(k-2)}, \quad k > 2, \tag{2}$$

where m describes the local competition between the strategies [8]. The gain of one strategy is the loss of another and local competition between the same strategies results in zero. The expected payoff for the local competition g_1 and g_2 of aggregators and senders are obtained to be

$$g_1 = (1-x)m, \qquad g_2 = -xm, \tag{3}$$

respectively, where m is given in (2). The average payoff ϕ of two strategies becomes

$$\phi = x(f_1 + g_1) + (1-x)(f_2 + g_2). \tag{4}$$

From Eqs. (1), (3), and (4), we obtain the replicator equation on graphs [8] for $k > 2$ to be

$$\dot{x} = x(f_1 + g_1 - \phi).$$

Substituting $\dot{x} = 0$ yields three equilibria: $x = 0, 1$, and

$$x = \frac{b(k^2+k-3) - (c-s)(k^2+2k)}{b(k+3)(k-2)}, \quad k > 2. \tag{5}$$

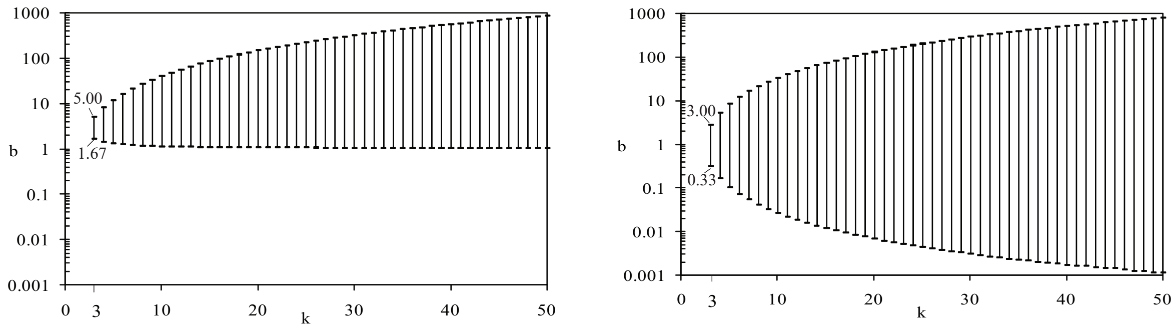
Note that the equilibria in (5) is feasible if $0 < x < 1$, i.e.,

$$\frac{k^2+2k}{k^2+k-3} < \frac{b}{c-s} < \frac{k^2+2k}{3}, \tag{6}$$

holds, where we use $c - s > 0$. We also have for all $k > 2$, $0 < (k^2+2k)/(k^2+k-3) < (k^2+2k)/3$. As a result, for any c, s, and k, there exists $b > 0$. Thus the equilibria in (5) is controllable and it can be shown to be stable.

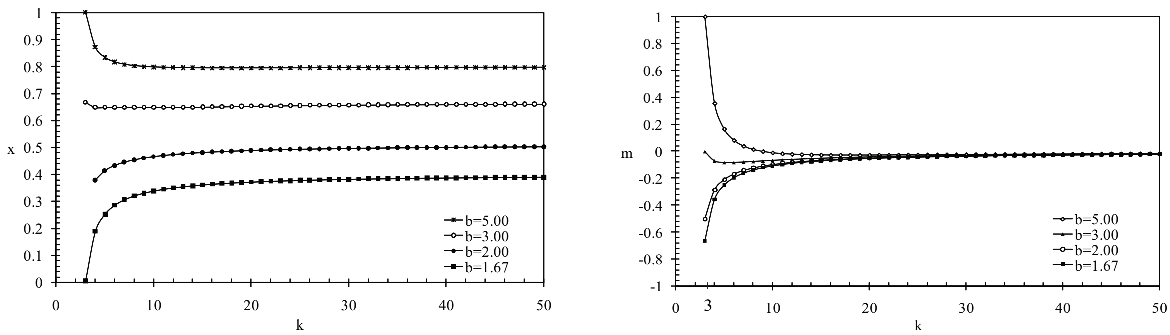
In case 2, we similarly obtain the stable and controllable equilibria to be

$$x' = \frac{b(k^2+k-3) - 3(c-s)}{(b+c-s)(k+3)(k-2)}, \quad k > 2, \tag{7}$$



(a) For case 1. (b) For case 2.

Figure 3. The supremum and infimum of b related with k for $c - s = 1$.



(a) Relation between *x* and *k* (case 1). (b) Characteristics of modifier *m* related with *k* for different *b* (case 1).

Figure 4. Effect of k on x.

for b such that

$$\frac{3}{k^2 + k - 3} < \frac{b}{c - s} < \frac{k^2 + k - 3}{3}. \qquad (8)$$

3.2 Numerical Results

We have four independent variables, b, c, s and k, which affect x. First, for simplification, assume that $c - s$ is equal to one. Note that this simplification does not lose generality. As a result, the ratio of aggregators can be controlled only by b and k according to Eq. (5) for case 1 and Eq. (7) for case 2. The expected number of aggregators can be obtained by the product of x and the number of cluster members.

Figs. 3(a) and 3(b) illustrate the relationship between k and the range of b with the supremum and infimum that satisfy Eqs. (6) for case 1 and Eq. (8) for case 2, respectively. In both cases, we observe that the valid range of b widens with k. The supremums for both cases are almost the same but the infimums for them present different characteristics: the infimum does not change for case 1 and decreases for case 2. As a result, the valid range of b for case 1 is about half of that for case 2. This is because as shown in Table 1, the fear to be a sender in case 2 is larger than that in case 1.

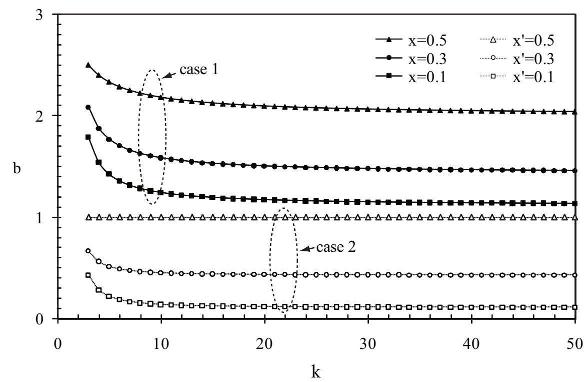


Figure 5. Relationship between k and b.

Next, we show how x varies according to the combination of b and k in Fig. 4(a) for case 1. x can be any value between 0 and 1, depending on b and k. For a specific b, x does not change when k becomes large. This is because m converges to zero with an increase of k as shown in Figs. 4(b). On the other hand, if k is less than 20, x shows different characteristics, depending on b. The smaller b is, the lower x is, and vice versa. This is because m becomes negative (positive) for small (large) b when k is less than 20 as shown in Eq. (2) and Fig 4(b). The similar results are also obtained for case 2.

Finally, we give appropriate combinations of b and k to achieve a given level of x, i.e., 0.1, 0.3, and 0.5 in Figs. 5 for case 1 and case 2. We first find that b can be less than 3.00 for case 1 and 1.00 for case 2. Note that this value of b is valid under the assumption of $c - s = 1$. We do not need much larger b to achieve our objective that is limiting the ratio of aggregators. Furthermore, if k is larger than 20, b converges to a value depending on the target level of x. Comparing two cases, we also find that case 2 can reduce b compared to case 1. In case 2, nodes prefer to be aggregators to avoid obtaining the lowest payoff.

4 Brief Summary of Simulation Results

Due to the limitation of space, we only show the brief summary of simulation results. The details of the results will be reported somewhere else. We evaluated our proposed scheme through simulation experiments by agent-based dynamics of evolutionary game theory [9]. The main goals of the agent-based simulation experiments are, a) to achieve the selection of the aggregators in an autonomous manner, b) to clarify the system scale valid for replicator dynamics, c) to investigate the convergence and transient properties of the aggregators selection procedure, and d) to investigate the role of each node in each round.

We first confirmed that the number of nodes in a cluster should be over 100 so that the prediction through replicator dynamics functions well. We observed that in each round we can obtain limited ratio x of aggregators which can be controlled by either parameter b or k. We also found that both cases almost approached the steady states at most 20 rounds (though the convergence time depends on the initial value of x). In addition, one interesting finding is that the role of each node is not fixed between two successive rounds. This feature may contribute to load balancing and robustness against node failures.

5 Conclusion

In this paper, we focused on reducing the visiting points of message ferries by aggregating all bundles with custody to some selected nodes. This will decrease the movement time and traveling cost of ferries as well as reduce the end-to-end delivery time of bundles. The node selection is conducted through decentralized processes with the help of strategic decisions of evolutionary game theory, where we can also control the numbers of the aggregators by proper parameters settings.

As a future work, we plan to solve the optimization problem of the message ferry's route between clusters. There are two main factors affecting the route optimization: ferry's transition time between clusters and a bundle generating rate in each cluster. The polling models in queueing theory [10] will help to solve this optimization problem by regarding each cluster or aggregator as a queue of bundles and the message ferry as a server.

Acknowledgment

This research was supported in part by "Global COE (Centers of Excellence) Program" of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

[1] V. Cerf and et al. Delay tolerant network architecture. Work-in-progress as an IETF RFC 4838 Draft. http://www.ietf.org/rfc/rfc4838.txt

[2] K. Fall. A delay-tolerant network architecture for challenged internets. In *Proc. of SIGCOMM*, pages 27–34, 2003.

[3] K. Fall and W. Hong. Custody transfer for reliable delivery in delay tolerant networks. Tech. report, in Delay Tolerant Networks, Intel Research, Berkeley-TR-03-030, 2003.

[4] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proc. of the Hawaii Int. Conf. on System Sciences*, page 10, 2000.

[5] M. A. Nowak. *Evolutionary dynamics: exploring the equations of life*. Harvard University Press, 2006.

[6] H. Ohtsuki, C. Hauert, E. Lieberman, and M. Nowak. A simple rule for the evolution of cooperation on graphs and social networks. *Nature*, 441(7092):502–505, 2006.

[7] H. Ohtsuki and M. Nowak. Evolutionary stability on graphs. *J. Theor. Biol.*, 251(4):698–707, April 2008.

[8] H. Ohtsuki and M. A. Nowak. The replicator equation on graphs. *J. Theor. Biol.*, 243(1):86–97, 2006.

[9] G. Szabó and G. Fáth. Evolutionary games on graphs. *Physics Reports*, 446(4-6):97–216, July 2007.

[10] H. Takagi. Queueing analysis of polling models: an update. *Stochastic Analysis of Computer and Communication Systems*, pages 267–318, 1990.

[11] J. Weibull. *Evolutionary game theory*. MIT Press, 1995.

[12] W. Zhao and M. Ammar. Message ferrying: proactive routing in highly-partitioned wireless ad hoc networks. In *Proc. of IEEE Future Trends of Distributed Computing Systems*, 2003.

[13] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *Proc. of ACM MobiHoc*, 2004.