Self-Organized Data Aggregation among Selfish Nodes in an Isolated Cluster

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. This paper considers a delay tolerant network, where a message ferry travels multiple isolated clusters, collects data from nodes in the clusters, and finally delivers the data to a sink node. In our previous work, we proposed a self-organized data aggregation technique for collecting data from nodes efficiently, which can automatically accumulate data from cluster members to a limited number of cluster members called aggregators. The proposed scheme was developed based on the evolutionary game theoretic approach, in order to take account of the inherent selfishness of the nodes for saving their own battery life. The number of aggregators can be controlled to a desired value by adjusting the energy that the message ferry supplies to the aggregators. In this paper, we further examine the proposed system in terms of success of data transmission and system survivability. We first introduce a new type of game model with retransmissions. Through both theoretic and simulation approaches, we then reveal feasible parameter settings which can achieve a system with desirable characteristics: Stability, survival, and successful data transfer.

Key words: delay tolerant networks (DTNs), evolutionary game theory, self-organized, aggregators, message ferry.

1 Introduction

In ambient information society, it is expected that each user can automatically obtain its desired information from environments equipped with a numerous number of devices. The underlying network supporting the ambient information society can be regarded as a kind of delay tolerant networks (DTNs) [1, 2] due to lack of reliable continuous end-to-end connectivity. In DTNs, a *store-carry-forward* [1] message delivery scheme and *custody transfer* [3] mechanism are used to confirm reliable transfer of data (bundle) with custody among nodes (devices) by delegating the responsibility of custody-bundle transfer through intermediate nodes in a hop-by-hop manner. The intermediate nodes keeping custody bundles are called *custodians*. Each custodian must reserve a sufficient amount of storage and energy to receive and hold the custody bundles until successful delivery or delivery expiration of the custody bundles. Due to the lack of the storages, custodians sometimes face storage congestion where they must refuse to receive any custody bundle from other nodes. In addition, each battery powered node must be awake while holding the bundles. Since each custodian also generates its own custody bundles, it is naturally selfish in behavior and rejects requests of custody transfer from other nodes to save its storage as well as its energy. Intuitively, this problem increases in long-term isolated networks.

In such a situation, some movable vehicles referred to as *message ferries* [9, 10] can solve the storage congestion problem by actively visiting the network and gather bundles from the custodians. Note that the message ferry has a sufficient amount of storages and energy to carry the bundles to the corresponding destination, i.e., a base station referred to as *sink node*, and it can also supply energy to the nodes if required. When there are several isolated networks referred to as *clusters*, the message ferry must visit each of the cluster and collects bundles from the custodians.

In such kind of scenarios, however, sometimes it is difficult for the message ferry to visit all of the nodes in a certain period of time. Taking account of the challenges, we developed a self-organized data aggregation technique in [4]. With the help of the evolutionary game theoretic approach [6, 7, 8], our system can automatically select some special custodians referred to as *aggregators*, which are cooperative in nature and willingly hold custody bundles of other nodes referred to as *senders*. Therefore, the message ferry needs to collect the bundles only from the aggregators. Note here that in this scheme, each aggregator must keep awake to receive and hold the bundles until transferring them to the message ferry, while each sender awakes only when generating and sending the bundles. In addition, each aggregator can obtain energy supply from the message ferry only when it finds a sender as its neighbor. In our scheme, each node appropriately selects strategy, i.e., sending or aggregating, depending on neighbors' strategies. This interaction among nodes is modeled as a game in game theory. The detail will be given in succeeding sections.

In this paper, we further examine the characteristics of the proposed scheme by focusing on unevaluated viewpoints in our previous work. We first introduce a new type of game model taking account of bundle retransmission when a sender cannot find an aggregator as its neighbor. Then, we evaluate the system stability through theoretic analysis based on replicator dynamics. Since the replicator dynamics only focuses on the strategy distribution, we further consider condition for system survivability. To grasp the node-level behavior, we also apply agent-based dynamics which is a simulation-based approach. Through simulation experiments, we evaluate the validity of the theoretic analysis and reveal feasible parameter settings to achieve successful bundle transfer.

The rest of the paper is organized as follows. We introduce our self-organized data aggregation scheme in section 2. Section 3 gives theoretic analysis of the system dynamics and the stable condition with the help of replicator equation on graphs. We also discuss the system survivability in section 3. After a brief introduction of agent-based dynamics, we evaluate the validity of the theoretic analysis and reveal feasible parameter settings achieving high successful prob-

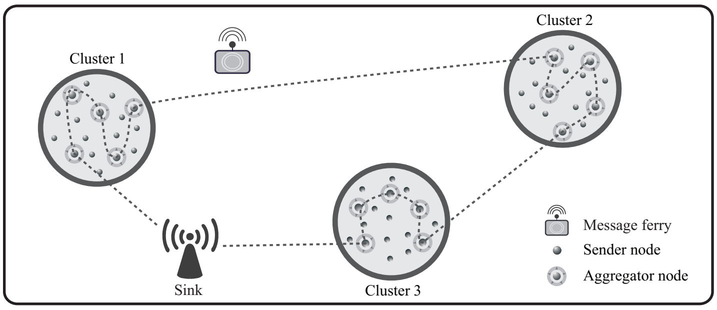


Fig. 1. Model scenario: Message ferry visits a limited number of aggregators in each cluster and delivers collected bundles to the sink node.

ability of bundle transfer through simulation experiments in section 4. Finally, section 5 concludes this paper.

2 Self-Organized Data Aggregation

We assume that a fixed sink node collects bundles from nodes in isolated clusters with the help of the message ferry as shown in Fig. 1. Each node in a cluster can directly/indirectly communicate with other cluster members (neighbors) within the transmission range but cannot communicate with the sink node and/or nodes in other clusters due to the long distances among the clusters. The message ferry serves the inter-cluster communication and visits only a limited number of aggregators in each cluster.

In this paper, we focus on bundle aggregation in a cluster. The bundle aggregation is conducted through three phases: a) Aggregator selecting phase, b) bundle aggregating phase, and c) bundle collecting phase. These three phases are repeated at each node and the unit of the repetition is referred to as *round*. Initially, each node randomly chooses to be an aggregator or a sender because it cannot know the neighbors' behavior. In the succeeding rounds, each node selects their role depending on the results of the previous round with the help of evolutionary game theory. During bundle aggregating phase, each sender transmits its bundles to one of the aggregators within the transmission range. Then, in bundle collecting phase, each aggregator allows to transmit its service request to the message ferry, transfers all bundles, and obtains energy supply from the ferry.

Due to the lack of reliable connectivity among arbitrary nodes, it is difficult to achieve a centralized control in DTNs. Therefore, the selection of aggregators should be performed in a decentralized way. Note that each node communicates only with its neighbors within the transmission range and is synchronized with each other. It determines to be an aggregator or a sender by mutual interaction based on its own benefit depending on the surrounding conditions. Since aggregators are always awake during a round while senders only wake up when generating and transmitting their bundles, it is assumed that each node loses

Table 1. Abstracted payoff matrix.

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

Table 2. Payoff matrix for no-retransmission case.

node 1 \ node 2	send	aggregate
send	*-s, -s*	*-s, b-c*
aggregate	*b-c, -s*	*-c, -c*

Table 3. Payoff matrix for retransmission case.

node 1 \ node 2	send	aggregate
send	*-c, -c*	*-s, b-c*
aggregate	*b-c, -s*	*-c, -c*

energy proportional to the length of time it keeps awake. Let c and s denote the amount of energy consumption for aggregators and senders, respectively, per round. s increases with retransmissions times but never exceeds c, i.e., $c \geq s > 0$. The energy supplied by the message ferry to each aggregator is represented by b. Intuitively, the larger b is, the more the aggregators increases. $b > c$ should be satisfied to keep battery of nodes alive.

The interaction among nodes is modeled as a game between two neighboring nodes in evolutionary game theory and is summarized as a payoff matrix. 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 where each node obtains different payoff from each combination of strategy. Table 1 illustrates the abstracted payoff matrix while Tables 2 and 3 illustrate the payoff matrices for no-retransmission case and retransmission case, respectively. Note that no-retransmission case is same as that proposed in our previous work [4]. The resulting payoffs for each combination 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 $P{=}c$ without any energy supply from the message ferry, because they are not be able to collect a sufficient number of bundles to request the message ferry to visit. An aggregator paired with a sender obtains the largest energy $T{=}b{-}c$; it loses c but obtains b from the message ferry, while the corresponding sender loses the smallest energy $S{=}s$. When both nodes select to be senders, two possible cases can take place depending on the presence of retransmission. For retransmission case, both of the senders consume $R{=}c$. This is equivalent to the worst case where each sender spends all the period of a round on achieving successful bundle transfer using retransmission mechanism. Note that we assume failure of bundle transfer is mainly caused by mismatch of waking time of sending and receiving nodes.

We obtain $T > S > R = P$ and $T > S = R > P$ for retransmission case and no-retransmission case, respectively. 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. On the other hand, condition $T > R$ and $S > P$ indicate that 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 [6]. 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.

3 Theoretical Analysis

In this section, we first analyse the relationship between the ratio of the number of aggregators and the parameters of the payoff matrix through replicator dynamics of evolutionary game theory on graph [6, 7, 8]. 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, and 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 DTNs. Moreover, we discuss a condition for the system survivability, under which each node can permanently be alive without battery shortage. Finally, some numerical results will be given.

3.1 Replicator Equation on Graphs

First, we introduce the replicator equation on graphs [7] for no-retransmission case, which was originally obtained in [4]. Let x_1 denote the ratio of the number of aggregators to the total number of cluster members for no-retransmission case. Note that $1 - x_1$ represents the ratio of the number of senders. Let k denote the number of neighbors of each node, called degree. For non-regular graphs, k represents the average degree. With the help of the payoff matrix in Table 2, the replicator equation on graphs for no-retransmission case becomes

$$\dot{x}_1 = x_1(1 - x_1)\left[\frac{b(k^2 + k - 3) - (c - s)(k^2 + 2k)}{(k + 3)(k - 2)} - bx_1\right], \quad k > 2.$$

Substituting $\dot{x}_1 = 0$ yields three equilibria: $x_1^* = 0, 1$, and

$$x_1^* = \frac{b(k^2 + k - 3) - (c - s)(k^2 + 2k)}{b(k + 3)(k - 2)}, \quad k > 2. \tag{1}$$

Note that the equilibrium in Eq. (1) is feasible if $0 < x_1^* < 1$, i.e.,

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

satisfies. We have for all $k > 2$, $0 < (k^2+2k)/(k^2+k-3) < (k^2+2k)/3$. Also $c-s > 0$ always holds. As a result, for any c, s, and k, there exists $b > 0$ which satisfies Eq. (2). Thus the equilibrium in Eq. (1) is controllable. Further, x_1^* in Eq. (1) is stable because $\dot{x}_1 > 0$ if $0 < x_1 < x_1^*$, and otherwise, $\dot{x}_1 < 0$.

Similarly, for retransmission case, with the help of the payoff matrix in Table 3, the stable and controllable equilibrium becomes

$$x_2^* = \frac{b(k^2+k-3)-3(c-s)}{(b+c-s)(k+3)(k-2)}, \quad k > 2,$$

which is valid for

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

In what follows, we call Eqs. (2) and (3) as *stable conditions.*

Note that at the equilibrium the ratio of the number of aggregators is fixed but the role (strategy) of each node may change [4]. This feature is suitable for our system such that each node can acquire opportunities to send bundles and obtain energy supply by changing its role (strategy) round by round.

3.2 Valid Parameter Settings for Permanently Alive System

Although we mentioned that each node has a chance to obtain energy supply by changing its role round by round, careful parameter tuning is required to achieve high system survivability. At the equilibrium, it is expected that each sender (aggregator) can find at least one aggregator (resp. sender) as its neighbor. Thus, expected payoff for each node becomes $\mathrm{E}[p] = (b-c)x_i^* - s(1-x_i^*)$ $(i = 1, 2)$. If $\mathrm{E}[p]$ is positive, the system could survive without loss of any node. Therefore, the valid combinations of b, c and s should satisfy the following condition for positive payoff (referred to as *running condition*):

$$\mathrm{E}[p] = (b-c)x_i^* - s(1-x_i^*) > 0 \quad (i = 1, 2). \tag{4}$$

In practice, the sink node tries to find appropriate x_i^* $(i = 1, 2)$ which satisfies both the stability condition and the running condition. The amount of energy supply from the message ferry, b, can be fully controlled by the sink node while c and s seem to be partly controllable: They are proportional to the length of waking period. The average node degree, k, is given from the environment. As a result, the sink node achieve desirable x_i^* $(i = 1, 2)$ by mainly controlling b. In the next subsection, we show some numerical results to illustrate the feasible parameter settings.

3.3 Numerical Results

In this section, we show some numerical examples of the adequate parameter settings satisfying the stable and/or running conditions according to the theoretic analysis in sections 3.1 and 3.2. First, we clarify the impact of stable and/or running conditions and the effect of retransmission mechanism. Fig. 2 depicts the valid range of controllable benefit b as a function of k when $c = 10$ and $s = 0.1$. Fig. 3 illustrates the corresponding range of x_i^* $(i = 1, 2)$. Note that we

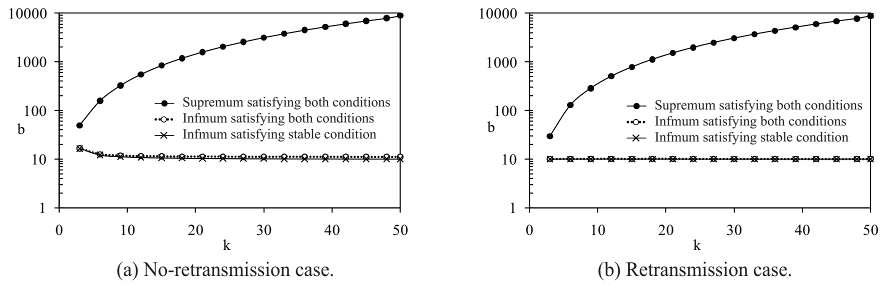


Fig. 2. The supremum and infimum of b satisfying stable and/or running conditions (c=10, s=0.1).

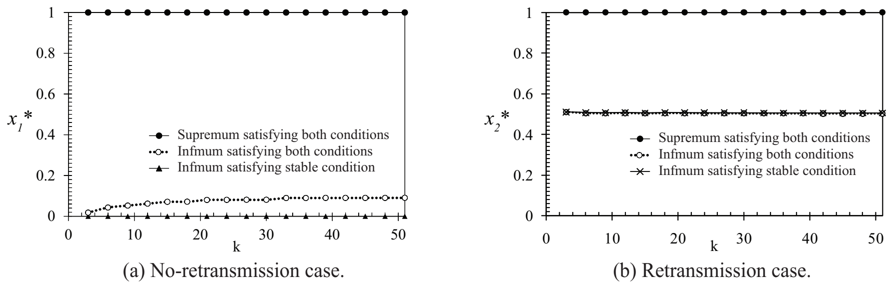
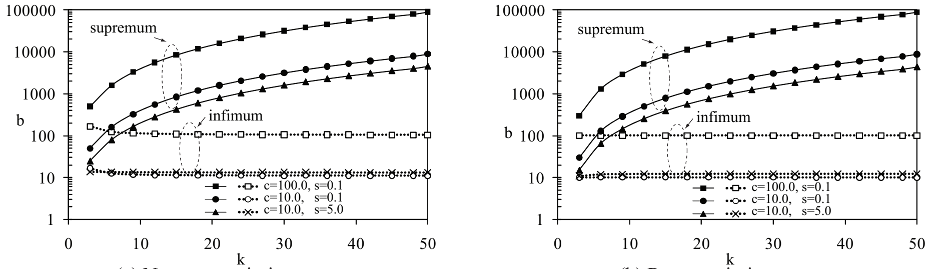


Fig. 3. The supremum and infimum of x_i^* (i=1,2) satisfying stable and/or running conditions (c=10, s=0.1).

show the results for larger k to reveal the basic characteristics even though they rarely occur in actual situations. We observe that the supremum of b increases with k and its has the same characteristic for both conditions. On the contrary, the infimum is almost constant while satisfying the two conditions. Although the presence of retransmission does not almost affect the valid range of b, Fig. 3 indicates that the retransmission mechanism narrows the valid range of equilibrium compared with no-retransmission case. Specifically, x_2^* must be greater than 0.505 to satisfy both conditions in retransmission case.

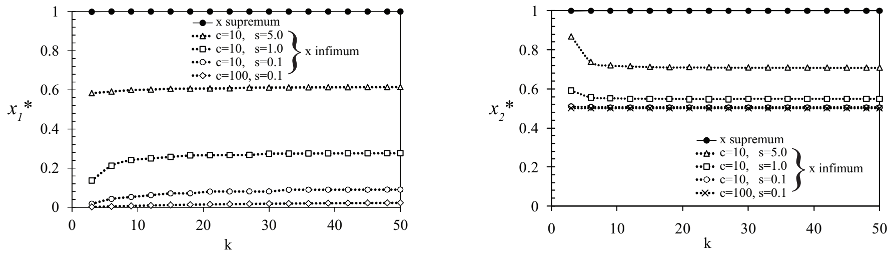
Next, we reveal how c and s affect the valid range of b and x_i^* $(i = 1, 2)$. Fig. 4 illustrates the supremum and infimum of b satisfying stable and/or running conditions when c and s vary. We observe that for a specific k, the range of b shifts up with c. This simply means that $b - c$ should be positive. On the contrary, increase of s decreases the supremum of b. This is because when senders lose more energy, temptation b to become an aggregator can be smaller.

Fig. 5 presents the supremum and infimum of x_i^* $(i = 1, 2)$ corresponding to Fig. 4. We observe that s has more impact on infimum than c. This is mainly caused by running condition. From Eq. (4), keeping low energy consumption of a sender is important for prolonging the battery life. We also find that no-retransmission case has wider feasible area than that with retransmission.



(a) No-retransmission case. (b) Retransmission case.

Fig. 4. The supremum and infimum of b satisfying stable and/or running conditions.



(a) No-retransmission case. (b) Retransmission case.

Fig. 5. The supremum and infimum of x_i^* $(i = 1, 2)$ satisfying stable and/or running conditions.

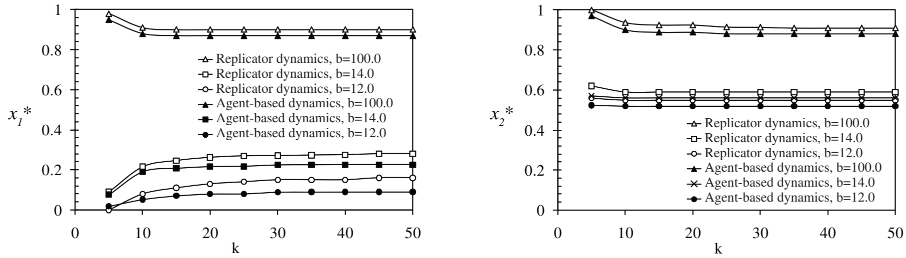
4 Simulation Experiments by Agent-Based Dynamics

4.1 Agent-Based Dynamics

In evolutionary game theory, the replicator dynamics predicts the macro-level system behavior which explains the effects of the corresponding parameters behaviors. On the other hand, the complementary method: The agent-based dynamics is used to understand the micro-level system behavior of the evolutionary game theory. It explains that with mutual interactions among neighboring nodes a superior strategy spreads over the network in a hop-by-hop manner.

In agent-based dynamics, in every round, each agent, i.e., node, first interacts with neighboring nodes and determines its strategy for the next round by comparing its own payoff with that of a randomly chosen neighboring node. The strategy update of node is conducted in a probabilistic manner where the more a strategy acquires the payoff, the more it spreads over the network through the imitation process in a hop-by-hop manner [4].

In what follows, we conduct simulation experiments for two purposes: 1) validation of analytical results, and 2) evaluations which are derived from micro-level system behavior.



(a) No-retransmission case. (b) Retransmission case.

Fig. 6. Validity of the theoretic analysis ($c = 10$, $s = 0.1$, $k = 5$).

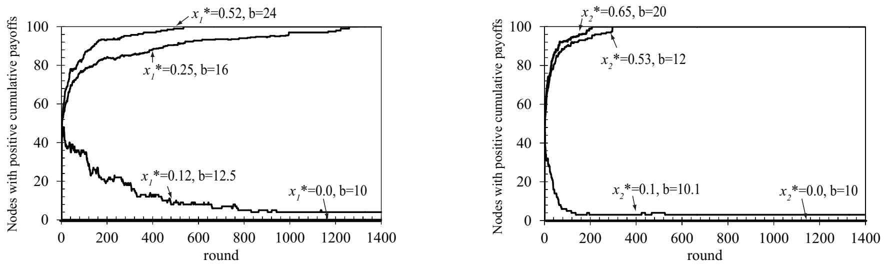
4.2 Simulation Models

The simulation experiments were conducted with a multi-agent programmable modeling simulator NetLogo [5] over unit disk graphs. The unit disk graphs are suitable for abstracting wireless networks because they are generated by randomly located nodes in 2-dimensional space where two nodes are adjacent if the distance between them is equal or less than a certain threshold, i.e., transmission range of each node. We set the number N of nodes to 100. The area size was set to 1 x 1 [km^2], and the transmission range of each node was set to 100 [m] in default. We controlled the average degree k by adjusting the transmission range of each node adequately. We assumed that the duration of a round was fixed and each node periodically generated a fixed number of bundles per round. Therefore, for simplicity, we set $c = 10$ and $s = 0.1$ in our simulation experiments. In the following figures, the average of 100 independent simulation experiments are plotted.

4.3 Validity of the Theoretic Analysis

Fig. 6 illustrates both analytical and simulation results for three examples selected from the valid parameter settings satisfying both stable and running conditions in the analysis. We observe that the analytical results slightly differ from the simulation results in both cases. These differences come from the relatively small system scale ($N = 100$) and the variance of node degree in the unit disk graphs. Since these characteristics naturally exist in the real networks, we can conclude that the analytical results can predict the system behavior with a certain degree of accuracy.

Next, we evaluate the validity of running condition, Eq. (4). As discussed in Section 3.2, all nodes can survive forever under appropriate values of the parameters b, c, and s. Fig. 7 shows the transition of the number of nodes with positive cumulative payoffs for different x_i^* ($i = 1, 2$). Note that every node initially has no payoff. Each node obtains energy supply from the message ferry only when it acts as aggregator and has at least one sender as its neighbor. Our aim is to achieve all nodes having positive cumulative payoffs such that



(a) No-retransmission case (The minimum value of b satisfying E[p] > 0 is 12.09).

(b) Retransmission case (The minimum value of b satisfying E[p] > 0 is 10.02).

Fig. 7. Transition of the number of nodes with positive cumulative payoffs ($c = 10$, $s = 0.1$, k=5).

they can work permanently if they have a sufficient amount of initial battery which depends on the parameter settings. Given $c = 10, s = 0.1, k = 5$, through theoretic analysis, minimum b, $b_{\min}$, satisfying Eq. (4), becomes 12.09 for no-retransmission case and 10.02 for retransmission case, respectively. As shown in Fig. 7, we observe that these theoretical predictions approximately accord with the simulation results except for b close to $b_{\min}$.

4.4 Successful Bundle Transfer

For senders (aggregators), it is desirable that at least one aggregator (resp. sender) exists as a neighbor for successful bundles transferring. We define the probability of senders (aggregators) that have at least one aggregator (resp. sender) in their neighbors as *sender (resp. aggregator) success probability.* These probabilities are affected not only by b, c, and s but also by k. Fig. 8 depicts the probabilities as a function of k. At first, Figs. 8 (a) and 8 (b) show that as k increases, the probabilities almost become one. This is because each node has more neighbors in average if k increases. Note that we omit the sender successful probability in Fig. 8(a) because senders always success in bundle transfer independent of neighbors' strategies with the help of the retransmission mechanism.

Next, comparing Figs. 8 (b) and 8 (c), we observe that small x_1^* decreases the probabilities when k is small. Note that we want to keep x_1^* relatively low as mentioned above but these results indicates that small x_1^* and k do not necessarily satisfy the probabilities close to one. To clarify this, Fig. 9 illustrates the minimum k which satisfies both probabilities ≥ 0.9 as a function of x_i^* $(i = 1, 2)$ where b is set adequately. We observe that the minimum k increases when x_i^* $(i = 1, 2)$ decrease but is kept relatively low. This can be confirmed from the fact that k should be greater than $1/x_i^*$ $(i = 1, 2)$ for senders to have at least one aggregator in their neighbors.

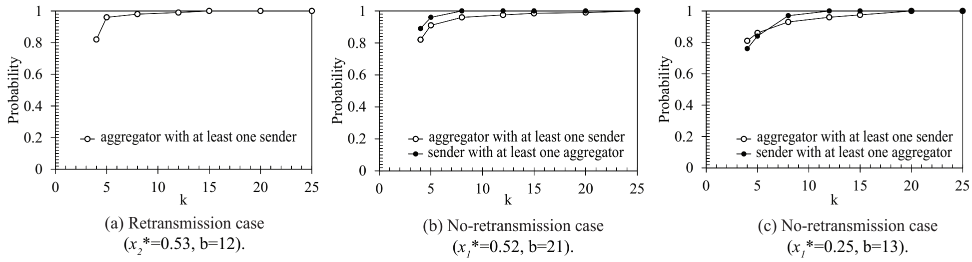


Fig. 8. Sender and aggregator successful probabilities satisfying stable and running conditions ($c = 10$, $s = 0.1$).

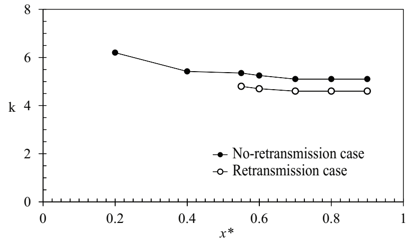


Fig. 9. Minimum k which satisfies sender and aggregator successful probabilities equal or greater than 0.9, and stable and running conditions ($c = 10$, $s = 0.1$).

5 Conclusion

In this paper, we further examined the characteristics of the self-organized data aggregation scheme proposed in [4]. We first introduced a new game model taking account of bundle retransmission when a sender cannot find an aggregator as its neighbor. Then, we derived the stable conditions through theoretic analysis based on replicator dynamics on graphs. In addition, we discussed running condition where all nodes can survive without battery outage. To evaluate the validity of theoretic analysis and reveal feasible parameter settings achieving successful bundle transfer, we conducted simulation experiments using agent-based dynamics. Both theoretic and simulation results presented appropriate parameter settings to achieve a system with desirable characteristics: Stability, survivability, and success probability in bundle transfer.

Acknowledgment

This research was supported by "Global COE (Centers of Excellence) Program" of the Ministry of Education, Culture, Sports, Science and Technology, Japan, as a part of "Ambient Networking" research group, Grant-in-Aid for Young Scientists (B) of JSPS under Grant No. 22700070, and Grant-in-Aid for Scientific Research (C) of JSPS under Grant No. 22500056.

References

1. Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., Weiss, H.: Delay tolerant network architecture. Work in progress as an IETF RFC 4838 draft. http://www.ietf.org/rfc/rfc4838.txt
2. Fall, K.: A delay-tolerant network architecture for challenged internets. In. ACM SIGCOMM, pp. 27–34 (2003)
3. Fall, K., Hong, W.: Custody transfer for reliable delivery in delay tolerant networks. Tech. Rep. IRB-TR-03–030, Intel Research Berkeley (2003)
4. Kabir, K.H., Sasabe, M., Takine, T.: Evolutionary game theoretic approach to self-organized data aggregation in delay tolerant networks. IEICE Transactions on Communications, vol. E93-B, no. 3, pp. 490–500, March (2010)
5. NetLogo, version 4.1, http://ccl.northwestern.edu/netlogo
6. Nowak, M.A.: Evolutionary dynamics: exploring the equations of life. Harvard University Press (2006)
7. Ohtsuki, H., Nowak, M.A.: The replicator equation on graphs. Journal of Theoretical Biology, vol. 243, no. 1, pp. 86–97 (2006)
8. Ohtsuki, H., Hauert, C., Lieberman, E., Nowak, M.A.: A simple rule for the evolution of cooperation on graphs. Nature, vol. 441, no. 7092, pp. 502–505 (2006)
9. Zhao, W., Ammar, M., Zegura, E.: A message ferrying approach for data delivery in sparse mobile ad hoc networks. In. ACM MobiHoc, pp.187–198 (2004)
10. Zhao, W., Ammar, M., Zegura, E.: Controlling the mobility of multiple data transport ferries in a delay-tolerant network.: In. IEEE INFOCOM, pp.1407–1418 (2005)