A Partially Centralized Messaging Control Scheme Using Star Topology in Delay and Disruption Tolerant Networks

Citation:

Shoki Oiyama, Hiroki Nishiyama, and Nei Kato, "A Partially Centralized Messaging Control Scheme Using Star Topology in Delay and Disruption Tolerant Networks," 2014 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), Monterey, USA, Jul. 2014.

A Partially Centralized Messaging Control Scheme Using Star Topology in Delay and Disruption Tolerant Networks

Shoki Oiyama, Hiroki Nishiyama, and Nei Kato
Graduate School of Information Sciences, Tohoku University, Sendai, Japan
E-mails: {oiyama, bigtree, kato}@it.ecei.tohoku.ac.jp

***Abstract*—Terminal-to-terminal communication technology has been studied to make communication and information distribution possible in an area, where communication infrastructure has been damaged or is not present, such as disaster area or rural area. Since mobile communication devices have short transmission range and high mobility owing to it bringing held by human beings, the network consisting of those nodes suffers from challenging conditions such as intermittent connectivity and absence of the end-to-end path between source and destination. To conquer that, Delay and Disruption Tolerant Networks (DTNs) using store-and-forward approach has attracted much research attention. In general routing algorithms, DTNs nodes exchange the control messages, e.g. the summary vector messages and request messages to know which messages each node possesses and to request replicated messages. However, the number of control messages being exchanged increases with the increase of node density. Furthermore, the size of a control message is proportional according to the number of replicas that shows. To reduce the number of control messages, we propose a partially centralized control scheme using star topology networks. Through our simulations, we show that our proposed method dramatically reduces control overhead in high node density scenarios.**

I. Introduction

In March 2011, the Great East Japan Earthquake struck and the communications infrastructures suffered immense damage. In that time, people could not use communications services through base stations to reach their family and friends. To achieve communication among acquaintances and information dissemination in disaster stuck areas, terminal-to-terminal communication technology has received significant research attention. Terminals in this kind of communication scheme are equipped with wireless communications technologies, including IEEE 802.11 and Bluetooth, to allow for communication between terminals. However, the transmission range of these wireless communications technologies is short and the terminals themselves exhibit high mobility due to user behavior. According to the aforementioned reasons, it is reasonable that the links between these terminals are frequently disrupted and network consisting of these terminals is that of opportunistic communications. Thus, we focus on Delay and Disruption Tolerant Networks (DTNs) [1] in this paper. DTNs use the store-and-forward approach to conquer the challenging conditions, which include intermittent connectivity and the absence of an end-to-end path between source and destination. In this network, a node replicates messages and forwards them to several nodes that it encounters to make use of redundancy. Furthermore, a node stores received messages in its buffer to make use of its mobility to deliver them to the other nodes or the destination node. In this way, replicated messages are propagated in this network and eventually the messages can reach their destination nodes. DTNs architecture is used in different fields e.g. interplanetary networks [2]–[4].

Much research effort was focused on DTNs routing algorithms [5]–[7]. In these general routing algorithms, nodes exchange control messages to avoid transmitting duplicate or unnecessary replicas. In many case, nodes send their possessing messages lists and request messages before forwarding replicas. This enables each node to send or receive only required messages. However, the overhead caused by these control messages increases as the node density becomes higher [8]. In disaster areas, disaster victims need shelter and food. Therefore, it is reasonable to assume that people are gathering at the places like refuges. Thus, making the node density high. Furthermore, the size of a control message increases with the increase of the number of replicas that shows. Thus, it is important to simplify control and reduce control messages for smooth communication.

In situations with high node density, it is worth to consider an alternative interface as an effective method to improve communication efficiency [9]. In this paper, we adopt a star topology network, which is widely used in terminal-to-terminal communications technology e.g. Wi-Fi Direct and Bluetooth. In this network, a group consists of one group head and several clients that is in the transmission range of the group head. This network topology has an advantage in the sense that the cooperation of grouped nodes and centralized control of the group head. However, a grouped node can not communicate with other group nodes and this network scale is limited by the transmission range of the group head. On this point, it is rational to say that many groups exist in the network and some users are moving around among those groups. Since, moving users keep moving around joining and leaving groups, this produces messages distribution among different groups. Figure 1 shows data propagation using star topology networks.

To reduce control messages, we propose a partially centralized control scheme using star topology network. In this scheme, group head manages message possessions list of clients and group communication. A group head can always

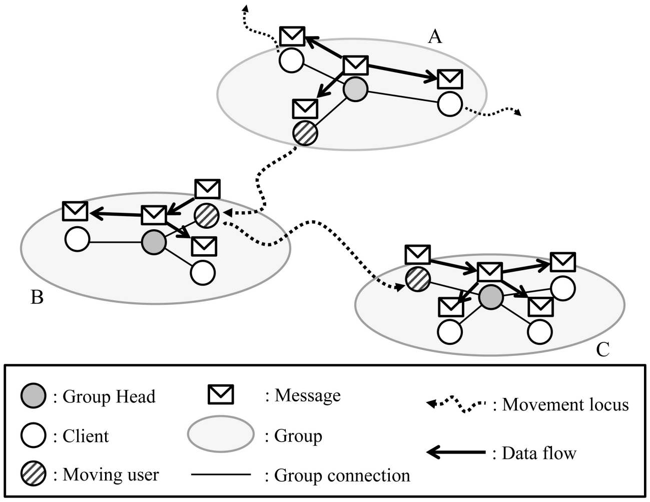

Fig. 1. Data propagation using a star topology network. The message is distributed to nodes leaving group A. Then, a node that left group A gets in contact and distributes the message to group B and C. Data is propagated to the whole network by the repetition of leaving and joining of these nodes.

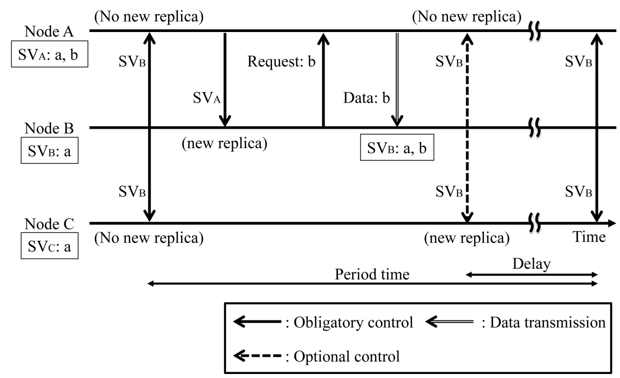

Fig. 2. Exchanges of control messages in DTNs routing algorithms.

recognize which messages client nodes possess and transmit replicas without request messages. As a result, this scheme reduces periodically transmitted possession lists and request messages from clients in the local area. In our proposal, we do not address how to organize and break up groups.

The rest of this paper is organized as follows. We show exchanges of control messages in the basic routing algorithms in Section II. Then, Section III provides analysis of control overhead. We propose a partially centralized control scheme that reduces control messages in Section IV. Section V evaluates the effectiveness of the proposed scheme. Finally, this paper is concluded in Section VI.

II. Message Exchanges in DTNs

Up to the present time, many DTNs routing algorithms have been proposed. In basic routing algorithms, nodes exchange control messages that show their knowledge or requirement apart from valuable data transmission. For example, epidemic routing uses information of what replicas each node possesses and wants [5]. Spray and Wait uses the aforementioned information in addition to L, which shows the remaining number of message replicas that can be propagated in the network [6]. PRoPHET especially uses information on the predictability values, which are calculated by history of encounters and transmissions [7]. As we have shown, DTNs nodes need a variety of information to propagate replicas.

Similar general routing algorithms, nodes exchange their possession list and request messages to avoid transmitting duplicate replicas. To investigate those, we consider epidemic routing, which is widely used and the most simple method among these routing algorithms. In epidemic routing, all nodes try to obtain replicas of messages that they do not possess and forward replicas that are requested to be sent by the neighbor nodes, to ensure high redundancy. This accomplishes a high delivery ratio and minimum delivery delay on condition that buffer capacity is enough to store all replicas and the transmission speed is high enough to transmit all replicas in an encounter. Simultaneously, nodes exchange Summary Vector (SV) messages and request messages as control messages. We show the procedure invoked when a node encounter occurs in Fig.2. First, each node broadcasts SV message which contains information on the replicas it buffers. A node that receives these messages can map location of surrounding replicas. Secondly, each node sends request messages that show replicas it requires based on information found in each SV message if necessary. These messages are transmitted by unicast to the node that transmitted SV messages. This prevents the transmission of duplicate replicas. Additionally, each nodes manages its own list of replicas it has requested to prevent duplicate requests. Finally, each node transmits the replicas it was requested. This procedure enables each nodes to send or receive only the required replicas. Besides this, SV messages are broadcasted periodically to report any changes in the possessions list that occured due to receiving or the creation of new messages [8]. However, the summary update takes a long time in case the period of time is too long. To shorten the update latency, each node broadcast updates not only periodically but also voluntarily in [10]. So we consider this control as an optional control. After that, a node that receives SV messages sends request messages for any replicas that it does not possess.

However, the number of control messages increases as the node density gets higher. This arises from the exponential increase of links in the network. Furthermore, control messages' size becomes larger according to the number of replicas they show. This is due to nature of the set data structure. In this way, these message exchanges are more complicated in that situation. Thus, it is important to simplify control and reduce control messages for smooth communication.

III. Analysis of Control Messages Overhead

In this section, we analyze control messages overhead with respect to two aspects, which is the number of control messages and their size in basic routing algorithms. Control overhead is calculated by multiplication of these. The number of control messages depends on node density as mentioned earlier. On the other hand, the data size of those depends on the data structure. To analyze control overhead, we consider the communication within an area with high node density e.g. refuges. We define the area to be consisting of n converging nodes that are within transmission range and can communicate with each other. Additionally, we define node mobility that a node comes into or goes out of this area. Node movement occurs at intervals of T_{m} in this area. However, we are interested in the influence of node density. To keep n cnstant with respect to the nodes movement, we assume one node enters and the other leaves simultaniously. Furthermore, each of the n nodes has unique data. We show our analysis of the number of control messages and the size of a control messages in the following subsections.

A. Number of Control Messages

In epidemic routing, nodes exchange SV messages and request messages as control messages. We show the number of these messages among n nodes in T seconds respectively in this subsection.

First, we calculate the number of SV messages. Ueda & Fujita, in their work in [8], assumed two types of exchange in SV messages. One is periodic broadcasting, in which each node transmits SV message every T_{p} seconds. Another is event-driven broadcasting, where each node sends SV message every time it is updated in addition to sending it periodically. In periodic broadcasting, each node broadcasts SV message every T_{p} seconds. Then, n nodes transmit SV message $\frac{T}{T_{\mathrm{p}}}$ times. Thus, the number of SV messages when periodic broadcasting is used, S_{p}, is given by:

$$S_{\mathrm{p}} = \frac{T}{T_{\mathrm{p}}} n. \tag{1}$$

In event-driven broadcasting, nodes additionally send SV messages when they generate new data or receive a replica. First, we show the number of replicas that are generated in T. A new message is generated in this area at intervals of T_{g}. Then, nodes in this area generate $\frac{T}{T_{\mathrm{g}}}$ new messages in T. Second, we show the number of replicas that are received in T. Initially, n nodes have messages. Therefore, the number of initial messages is n. Besides this, new messages are generated and brought in to this area by nodes with mobility. $\frac{T}{T_{\mathrm{g}}}$ messages are generated and $\frac{T}{T_{\mathrm{m}}}$ nodes bring a message finally. Thus, the number of all the replicas is shown by $n+\frac{T}{T_{\mathrm{m}}}+\frac{T}{T_{\mathrm{g}}}$. We assume that the transmitting speed of nodes is high enough to receive all replicas before they leave the area. However, nodes cannot receive replicas that are generated after left the area. Thinking of this, we consider to separate n nodes into $n-1$ stationary nodes and one moving node separately. Stationary nodes receive all replicas except the source node. Hence, the number of replicas stationary nodes received is $(n-2)(n+\frac{T}{T_{\mathrm{m}}}+\frac{T}{T_{\mathrm{g}}})$. On the other hand, the number of replicas that the moving node receives increases in proportion to time. Thus, we can derive average number of replicas as $(n+\frac{T}{2T_{\mathrm{m}}}+\frac{T}{2T_{\mathrm{g}}})$. Additionally, moving nodes include not only the new $\frac{T}{T_{\mathrm{m}}}$ nodes but also the initial node that have already moved out the area. In this way, the number of replicas moving nodes received is $(\frac{T}{T_{\mathrm{m}}}+1)(n+\frac{T}{2T_{\mathrm{m}}}+\frac{T}{2T_{\mathrm{g}}})$. Then, the number of SV messages when event-driven broadcasting is used, S_e, is given by:

$$\begin{aligned} S_{\mathrm{e}} =& S_{\mathrm{p}} + \frac{T}{T_{\mathrm{g}}} + (n-2)(n+\frac{T}{T_{\mathrm{m}}}+\frac{T}{T_{\mathrm{g}}}) \\ &+ (\frac{T}{T_{\mathrm{m}}}+1)(n+\frac{T}{2T_{\mathrm{m}}}+\frac{T}{2T_{\mathrm{g}}}). \end{aligned} \tag{2}$$

Secondly, we show the number of request messages. A node having received SV messages send request messages that show a list of replicas it requires [11]. In this time, each nodes manages its own list of replicas it has requested. This guarantees that a node does not request duplicate replicas and avoids useless transmissions. That means that a node sends request messages at the first time and the time received SV contains some new replicas caused by node movement or the generation of new messages. Then, the number of request messages is independent of the number of SV messages. Nodes send request messages to each other at first, which is counted as $n(n-1)$. Additionally, in node moving, a moved node sends request messages to $n-1$ nodes and $n-1$ nodes send those to a moved node, which is counted as $2\cdot\frac{T}{T_{\mathrm{m}}}(n-1)$. Besides these, nodes apart from the source node send request message when a new message is generated, which is counted as $\frac{T}{T_{\mathrm{g}}}(n-1)$. Thus, the number of request messages in basic routing, R_{b}, is given by:

$$R_{\mathrm{b}} = n(n-1) + 2\cdot\frac{T}{T_{\mathrm{m}}}(n-1) + \frac{T}{T_{\mathrm{g}}}(n-1). \tag{3}$$

According to (1), (2), and (3), the number of control messages obviously depends on the number of nodes. This indicates that message control is more complicated as the node density gets higher.

B. Size of Control Messages

In this subsection, we show the size of a control message to evaluate control messages overhead. Control messages exchanged among nodes shows the list of the possession and the request as previously mentioned. These messages use a set data structure and its data size generally increases with the increase of the number of elements. To evaluate control overhead of exchanges of control messages, we assume two types of data structures indicating a set, i.e., a simple array and a bloom filter. A simple array shows a set by enumerating elements. In this method, each node has a hash table, which stores data with a corresponding hash value, which is a constant-length bit array derived by a hash function. We define the hash value is 128bits long, which result in a space is equivalent to IPv6. This makes possible the description any element as practically independent. So, it is possible for a node to identify the replicas another node possesses by just exchanging the hash value lists. Then, the data size of a simple array is calculated by multiplication of a constant value and the number of elements.

On the other hand, the bloom filter is a space-efficient probabilistic data structure that answer queries whether an element is contained in a set or not [12]. Shukla *el al.*, in their work conducted in [13], used a bloom filter to reduce control messages overhead. To reduce data volume, the bloom filter has a common bit array for a set. This bit array maps each elements in it by using hash functions. This common bit array enables the overlap of bits between some elements. Therefore, the bloom filter has low data volume. However, it is possible to answer that the element is a member of the set in spite of it not being included in the set, i.e., false positives in this data structure. Tarkoma *el al.*, in their work in [12], showed this probability rate and we can derive that about 10 bits per element is needed to realize probability lower than 1%.

We show a comparison of these data structures in terms of only data size. The size of a simple array is proportional to the number of the elements, while that of a bloom filter is a constant value. This indicates that a simple array has lower volume compared to a bloom filter if the number of elements is small. Because a bloom filter contains useless bits to record more elements in that case. However, a simple array occupies a large volume compared to the bloom filter in case the number of elements increases. So, both of these data structures have a disadvantage, which makes control overhead higher. In conclusion of our analysis, we observe that it is important to reduce control messages for smooth communication.

IV. Centralized Control Scheme

Through our analysis, we find out that the number of control messages increases with the increase of node density. Furthermore, the size of a control message grows with the number of replicas that it shows. In this way, exchanges of those cause high overhead. To reduce control overhead, we aim to reduce the number of control messages. In this section, we propose a partially centralized control scheme using star topology networks. This network consists of one group head and several clients that are with in the transmission range of the group head. A group head forms a group and clients which are with its transmission range join that group. In this time, grouped nodes are restricted to be a part of one group at a time and can only communicate with nodes that are with in the same group. Nodes in a star topology network cooperate under the centralized control of the group head. In this scheme, group head manages possessions list of clients and controls group communication. Thus, a group head can always recognize which replicas client nodes possess. To realize this, a group head aims to collect all replicas and SV.

We show procedure on our proposal in Fig. 3. Our proposal has three phases, which are the summary grasping phase, the data collecting phase, and the data distributing phase. In summary gasping phase, each client node sends SV message to the group head. In this time, a group head can map all replicas location in the group. In data collecting phase, the group head sends request messages to some client nodes which possess messages it does not possess. Receiving these, clients transmit

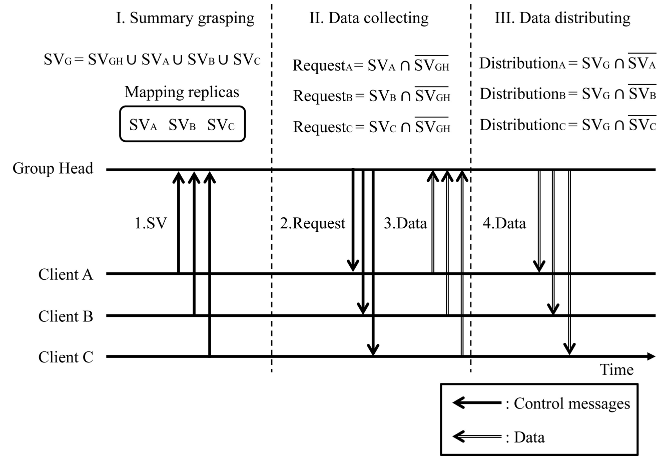

Fig. 3. Procedure of our proposal.

required replicas to the group head. This completes all replicas collection by the group head. In data distributing phase, group head distributes insufficient replicas to clients founded by possessions list. A group head has knowledge of replicas location, so it can easily find out which replicas to transmit to either client. Besides this, clients need to report summary updates to inform the group head of new information. It is impossible for a group head to find out that a message is generated in a client unless the client reports. So, when a message is generated in the node after sending SV message, it sends the updated SV. With that, the group head receives SV message and sends a request message to the corresponding client. In our scheme, a group head never sends SV message. Additionally, clients send these vector messages initially and only when new data is generated. Furthermore, clients never send request messages. In this way, our proposed method can decrease the control message overhead.

Now, we calculate the number of control messages in case of our proposal. We show the number of SV messages at first, then that of request messages. In our proposal, a group head collects clients' SV messages when a new node join in the group. Additionally, a client sends SV message when new data is generated or moves into the group. Then the number of SV messages in our scheme, S_{o}, is given by:

$$S_{\mathrm{o}} = (n-1) + \frac{T}{T_{\mathrm{g}}} + \frac{T}{T_{\mathrm{m}}}. \tag{4}$$

On the other hand, a group head sends request messages to the client if its SV contains new data. As we have shown, clients send SV messages at first or the time a new message is generated. Furthermore, we assume the case where each node has unique data. This means clients have new data whenever they send SV messages. Thus, the number of request messages in our scheme, R_{o}, is equal to the number of SV messages, i.e., $R_{\mathrm{o}} = S_{\mathrm{o}}$.

Up to here, we showed the simplification of control messaging in local areas without observation of the whole network. In our overview, many star topology networks exist in the

TABLE I
A LIST OF NOTATIONS DEFINED AND USED IN OUR ANALYSIS.

n	Number of nodes.
T_{p}	Period of broadcasting.
T_{m}	Average interval of moving.
T_{g}	Average interval of message generation.
S_{p}	Number of SV messages in periodic broadcasting.
S_{e}	Number of SV messages in event-driven broadcasting.
R_{b}	Number of request messages in basic routing.
S_{o}	Number of SV messages in our scheme.
R_{o}	Number of request messages in our scheme.

TABLE II
SIMULATION PARAMETERS

Simulation time	100s
Bandwidth	1Mbps
Data size of replicas	12.5kB
Period of broadcasting	5s
Average interval of moving	20s
Average number of new data per second	0.2
Number of trials	500

whole network and each of those uses our control scheme. We expect that much partial improvement is accumulated in the whole network and that it impacts efficiency of networking. In this scheme, the simplification of control in a group influences the number of transmittable replicas in constant time. Additionally, this also influences the propagation rate and speed of replication. Ultimately, it accomplishes high delivery ratio and short delay, which is the aim of DTNs.

Finally, our scheme indicates partially networking's potential to improve efficiency in various DTNs routing algorithms. In this scheme, we use the advantage that cooperation of grouped nodes and centralized control of the group head. Our proposed method can be applied to basic routing exchanging control messages to decrease control messaging overhead. A group head arranges these messages. Then, a group head distributes required information and manages replicas transmission. In this process, every client node only communicates with a group head. So, we expect that the efficiency of DTNs routing algorithm can be improved with our scheme. All the notations used in our analysis are summarized in Table I.

V. EXPERIMENT

In this section, we evaluate our proposal in two scenarios by simulation. We consider the communication within an area with high node density as we had shown in Section III. First, we evaluate it in terms of just control messages. Secondly, we evaluate it in terms of influence of replica distribution.

A. Effect on reduction of control messages

In this scenario, we show the effect of changing the number of nodes on the reduction of control messages. Each node has an unique replica, which never have destination in this area. Then we consider only the distribution of replicas. Additionally, we assume that nodes move and generate data according to a probabilistic distribution. We define the time between node movement following an exponential distribution and the number of generated data following a Poisson distribution. This simulation parameters are shown in Table II.

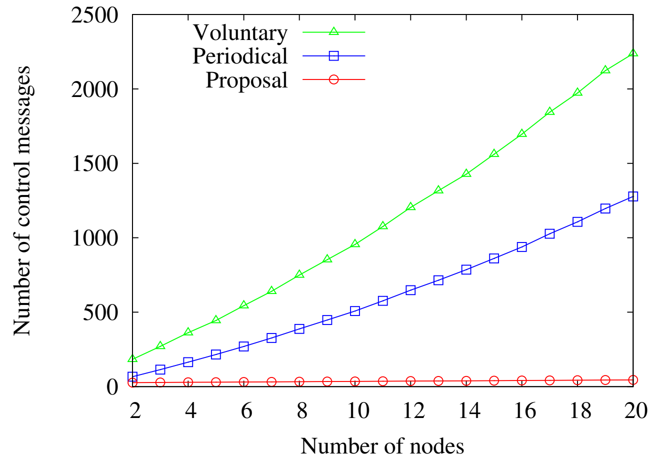

Fig. 4. Exchanges of control messages when the number of nodes increases.

Figure 4 shows the number of control messages in basic routing algorithms and our proposal. In basic routing algorithms, we find that the number of control messages dramatically increases as the number of nodes increases. On the other hand, the number of control messages in our proposal hardly increases. This is because our scheme aims to collect replicas to the group head in the beginning, while basic routing algorithms aim to enable any nodes to collect replicas. In our scheme, control messages are sent only at the required time to report the possession list or collect replicas to the group head. We see that our scheme can simplify control messaging in this situation because star topology network operates in local areas. On the other hand, in basic routing methods, SV messages are sent periodically or voluntarily because each nodes does not know the network topology and the data flow. Additionally, the request messages are sent by any node that aims to obtain replicas. In this way, control of replicas transmission is complicated.

B. Effect on distribution replicas

In this scenario, we evaluate our scheme in terms of influence of replica distribution in high density circumstances. To deliver messages to the destination node, a DTNs node has to distribute replicas to some extent. Therefore, we want to know how long it takes for a certain number of replicas to be distributed. To evaluate the influence of control messages, transmission of replicas and control messages are done on the same channel. In our simulation, the number of nodes is high, constant at a value of 20, and nodes uses a bloom filter to describe a set data structure. We assume that the inerrant static bloom filter is capable of 1000 elements, whose data size is 1.25kB. Additionally, each node has 10 unique replicas, which never have destination in the current area that the node is in.

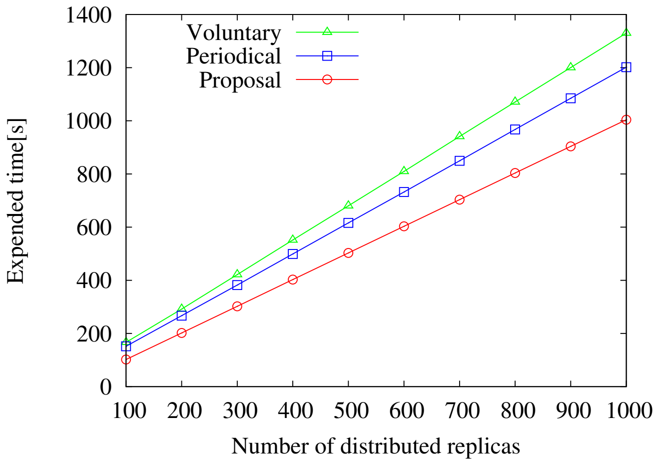

Fig. 5. Distributed replicas.

Then we consider only the distribution of replicas.

We show the simulation results in Fig. 5. We can see that the proposed scheme distributes replicas faster than basic routing algorithms. This indicates that control overhead can influence the replicas propagation speed. Then our scheme makes sense of simplification of complicated control and smoothing data propagation in the local area. Finally, we consider what kind of influence this effect produces in the whole network. We expect that smoothing propagation lets more users receive distributed data in a certain time. This is important for data propagation in the whole network because some moving users convey data among different areas. It is reasonable that data propagation rate is improved by the increase of the source of data distribution. This also accomplish high delivery ratio and short delay, which is the aim of DTNs.

VI. Conclusion

DTNs use the store-and-forward approach to conquer the challenging conditions of intermittent connectivity and absence of the end-to-end path between the source and the destination. In DTNs, a node replicates messages and forwards them to nodes that it encounters for the sake of redundancy. Up to the present time, many DTNs routing algorithms were proposed. In general routing algorithms, nodes exchange control messages such as SV messages and request messages to avoid transmitting duplicate replicas. The number of control messages increases with the increase of the number of nodes. Furthermore, the size of a control message grows with the number of replicas that it shows. To reduce the number of control messages, we propose a partially centralized control scheme using star topology networks. This network uses the cooperation of grouped nodes and the centralized control of the group head. Group heads can always recognize which messages client nodes posses by SV and history of transmission. Then, clients send those only when the group is formed or a new message is generated at themselves. Furthermore, clients never send requested messages. In this way, the number of control messages can be reduced. Through simulation, we derive that our proposal reduce the number of control messages dramatically. Additionally, it enable nodes to propagate replicas faster than basic routing. As a future of this work, we consider to adopt multicast to enhance our scheme and replicas distribution. A star topology network is suitable for multicast because a group head can communicate with every clients. However, we need to consider related problems like radio frequency interferences and error correction. We are going to clarify strong and weak point of multicast in star topology network.

Acknowledgment

This research was partially funded by the project, “Cognitive Security: A New Approach to Securing Future Large Scale and Distributed Mobile Applications,” of Japan-US Network Opportunity: R&D for ”Beyond Trillions of Objects” supported by National institute of Information and Communications Technology (NICT), Japan.

References

[1] K. Fall, “A Delay-tolerant Network Architecture for Challenged Internets,” in *the SIGCOMM 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Aug. 2003, pp. 27–34.

[2] C. Caini, H. Cruickshank, S. Farrell, and M. Marchese, “Delay- and disruption-tolerant networking (dtn): An alternative solution for future satellite networking applications,” *Proceedings of the IEEE*, vol. 99, no. 11, pp. 1980–1997, Nov 2011.

[3] I. Bisio, M. Cello, T. De Cola, and M. Marchese, “Combined congestion control and link selection strategies for delay tolerant interplanetary networks,” in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, Nov 2009, pp. 1–6.

[4] I. Bisio, M. Marchese, and T. De Cola, “Congestion aware routing strategies for dtn-based interplanetary networks,” in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, Nov 2008, pp. 1–5.

[5] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, “MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks,” in *25th IEEE International Conference on Computer Communications (INFOCOM 2006)*, Apr. 2006, pp. 1–11.

[6] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks,” in *the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking*, Aug. 2005, pp. 252–259.

[7] A. Lindgren, A. Doria, and O. Schelen, “Probabilistic Routing in Intermittently Connected Networks,” *Mobile Computing and Communications Review*, vol. 7, no. 3, pp. 19–20, Jul. 2003.

[8] H. Ueda and N. Fujita, “A Message Transmission Control Scheme for Efficient Information Sharing in Disruption Tolerant Networks,” in *Intelligent Networking and Collaborative Systems (INCOS)*, Nov. 2010, pp. 392–397.

[9] H. Nishiyama, M. Ito, and N. Kato, “Relay-by-Smartphone: Realizing Multihop Device-to-Device Communications,” *IEEE Communications Magazine*, pp. 2–11, Apr. 2014.

[10] M. Akdere, C. c. Bilgin, O. Gerdaneri, I. Korpeoglu, O. Ulusoy, and U. Çetintemel, “A Comparison of Epidemic Algorithms in Wireless Sensor Networks,” *Computer Communications*, vol. 29, no. 13-14, pp. 2450–2457, Aug. 2006.

[11] A. Vahdat and D. Becker, “Epidemic Routing for Partially-Connected Ad Hoc Networks,” Tech. Rep., 2000.

[12] S. Tarkoma, C. Rothenberg, and E. Lagerspetz, “Theory and Practice of Bloom Filters for Distributed Systems,” *IEEE Communications Surveys Tutorials*, vol. 14, no. 1, pp. 131–155, First Quarter 2012.

[13] S. Shukla, N. Kumar, and R. Misra, “Impact of Bloom Filter on Infection Rate in Epidemic Forwarding for ICNs,” *Wireless Personal Communications*, vol. 75, no. 4, pp. 2165–2180, Apr. 2014.