Studies on

Ferry-Assisted Multi-Cluster Delay Tolerant Networks

K. Habibul Kabir

Osaka University

2011

Studies on

Ferry-Assisted Multi-Cluster Delay Tolerant Networks

K. Habibul Kabir

Osaka University, Japan
Department of Information and Communications Technology
Division of Electronic and Information Engineering
Graduate School of Engineering

2011

Declaration

This dissertation is submitted to fulfill the requirements for the degree of Doctor of Philosophy in the Graduate School of Engineering at Osaka University, Japan.

Except where explicit reference is made, this dissertation is the result of my own work at Takine Laboratory, Department of Information and Communications Technology, Division of Electronic and Information Engineering, Graduate School of Engineering in Osaka University, Japan, and includes nothing which is the outcome of work done in collaboration with any other Institution. No part of this dissertation has been or is being submitted for a degree or diploma or other qualification at any other University.

K. Habibul Kabir

Osaka University
Osaka, Japan
2011

DEDICATED

TO MY FATHER

his continuous inspiration and courageous support have brought me this far . . .

Preface

DELAY tolerant networks (DTNs) provide networking in infrastructure-less environments, e.g., deep space, rural areas, disaster areas, underwater fields, etc. In DTNs, the current TCP/IP model cannot work well due to lack of continuous end-to-end connectivity. A store-carry-forward message delivery scheme and custody transfer mechanism is used in DTNs to confirm reliable transfer of bundles with custody transfer request among nodes, by delegating the responsibility of custody-bundle transfer through intermediate nodes in a hop-by-hop manner. Note that a bundle is the protocol data unit in DTNs. The intermediate nodes keeping custody bundles are called custodians. Each custodian must reserve a sufficient amount of storage and energy for receiving and holding the custody bundles until their successful delivery or delivery expiration. Due to shortage of storage capacity, custodians sometimes face storage congestion, where they have to refuse to receive any custody bundle from other nodes. In addition, each battery-powered node has to 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 is aggravated in long-term isolated networks.

In such a situation, some movable vehicles referred to as message ferries can solve the storage congestion problem by actively visiting the network and gather bundles from custodians. Note that message ferries are equipped with a storage enough to carry collected bundles to the destination, i.e., a base station referred to as a sink node, and it can also supply energy to the custodians if required. When there are multiple isolated networks referred to as clusters, message ferries have to periodically visit those clusters and collects bundles from custodians there. This network architecture is suitable for wide area sensing. Note that each node in a cluster can directly/indirectly communicate with other cluster members through multi-hop communication but cannot communicate with nodes in other clusters due to long distances among them. A message ferry helps the inter-cluster communication by

acting as a mediator between each cluster and the outer world via the sink node which serves as a connector to the Internet or to other sink nodes.

In such ferry-assisted multi-cluster scenario, inter-cluster communication and intra-cluster communication should be carefully considered to minimize the total mean delivery delay of bundles. This thesis mainly focuses on these inter-cluster and intra-cluster communications by combining three research studies. The inter-cluster communication has been studied by two studies: 1) Grouping clusters, and 2) Optimal visiting order of isolated clusters. And, the intra-cluster communication has been studied by self-organized data aggregation technique among selfish nodes in an isolated cluster.

Thesis Organization

The content of this thesis is organized into following five chapters:

1. Introduction
2. Self-organized data aggregation technique
3. Optimal visiting order of isolated clusters
4. Grouping clusters
5. Conclusions

Chapter 1 provides the background of recent development of DTNs, the motivation for the studies and the aim of the research.

Chapter 2 addresses intra-cluster communication by self-organized data aggregation technique among selfish nodes in an isolated cluster. We proposed a self-organized data aggregation technique for collecting data from nodes efficiently, which can automatically accumulate data from nodes in a cluster to a limited number of nodes (called aggregators) in the cluster. 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. We further examine the proposed system in terms of successful data transmission, system survivability and the optimality of aggregator selection. We introduce two game models by taking account of the retransmissions mechanism of bundles. Through both theoretical and simulation-based approaches, we reveal feasible parameter settings that can achieve a system with desirable characteristics of stability, survivability, and successful data transfer.

Chapter 3 focuses on one part of inter-cluster communication by studying the optimal visiting order of isolated clusters. When there are lots of distant static clusters, the message ferry should visit them efficiently to minimize the mean delivery delay of bundles. We propose an algorithm for determining the optimal visiting order of isolated static clusters in DTNs. We show that the minimization problem of the overall mean delivery delay in our system is reduced to that of the weighted mean waiting time in the conventional polling model. We then solve the problem with the help of an existing approach to the polling model and obtain a quasi-optimal balanced sequence representing the visiting order. Through numerical examples, we show that the proposed visiting order is effective when arrival rates at clusters and/or distances between clusters and the sink are heterogeneous.

Chapter 4 focuses on another part of the inter-cluster communication by studying the grouping of clusters. When there are lots of distant static clusters, multiple message ferries and sink nodes will be required. We aim to make groups each of which consists of physically close clusters, a sink node, and a message ferry. Our main objective is minimizing the overall mean delivery delay of bundles in consideration of both offered load of clusters and distance between clusters and their sink nodes. We first model this problem as a nonlinear integer programming, based on the knowledge obtained in our previous work. Because it might be hard to solve this problem directly, we take two-step optimization approach based on linear integer programming, which yields an approximate solution of the problem. Through numerical results, we show the two-step optimization approach works well.

Chapter 5 presents the conclusions of this thesis by summarizing all results and observations we obtained through the researches.

K. Habibul Kabir

Osaka University
Osaka, Japan
2011

Acknowledgements

I would like to appreciate many people who have made important contribution to this research work and who have made my life in Osaka University an enjoyable and unforgettable experience.

First and foremost, I would like to express my sincere appreciation to my supervisor, Professor Tetsuya Takine. His valuable inspiration, thoughtful and patient guidance and encouragement throughout the entire research, was probably the only thing that kept me going. He also created an indispensable environment for me to conduct my research work. He supervised me in writing all my papers and this thesis and took the painstaking job of correcting those, line by line, at a remarkable pace.

I would like to express my gratefulness to my thesis committee members, Professor Seiichi Sampei and Professor Takashi Washio for their intensive reading, insightful suggestions and constructive comments on this thesis.

I also take pleasure in thanking Professor Shozo Komaki, Professor Zenichiro Kawasaki, Professor Kenichi Kitayama, Professor Noboru Babaguchi, Professor Kyo Inoue and Professor Riichiro Mizoguchi of Osaka University. Their valuable discussions and favorable comments during my course of study and presentations, greatly enhanced the quality of this research.

I would like to thank Associate Professor Takahiro Matsuda for his support during my initial stage of the research. It was my great pleasure to have the opportunity to attend his seminars on wireless networking which helped me a lot to build up my basic knowledge.

I am highly indebted to Assistant Professor Masahiro Sasabe for his enthusiasm and stimulating discussions about both technical and non-technical topics. He helped me to plant the initial seed for my research and gave me enough freedom to pursue my own ideas. Researching in his group provided me the chance to gain insight into highly fascinating topics of delay tolerant networks.

I would like to thank Dr. Mutsunori Yagiura of Nagoya University for his valuable suggestions and comments regarding my study of grouping clusters. A very especial

thank to Dr. Hirantha Sithira Abeysekera for his many enlightening discussions and comments from my very first day in Takine Laboratory. All kinds of remarkable and friendly supports from him are really unforgettable.

My special appreciation goes to "Monbukagakusho" and "ICOM Electronics and Communication Engineering Foundation" for their funding supports which greatly helped me to carry out my research. This research was also supported by Global COE (Centers of Excellence) Program for "Founding Ambient Information Society Infrastructure," MEXT, Japan. I would like to thank all committee members of Global COE Program.

I am very grateful to Dr. Kouji Hirata, Ph.D candidates Takanori Kudo and Tomotaka Kimura for their encouragement and creditable suggestions. I am also deeply grateful to Ms. Sachiko Hashimoto and Ms. Yumi Shimoyashiki for their continuous supports. The friendly and hard-working faces of all other members of Takine Laboratory (past and present) surely kept me awake during numerous sleepy afternoons. They taught me how to be a true "Majime" and the inner meaning of "Gambaru".

I would like to thank my very good friend Dr. Tawfique Hasan Tazin and Dr. Salahuddin Zabir for their enlightening and inspirational talks. Especial thanks to all the Bangladeshis in Kansai area for their extraordinary and memorable supports.

I am grateful to my host family, Mr. and Mrs. Otusuki for their noteworthy cares. I would also like to express my significant thank to Mr. Shintaro Iefuji and Mrs. Noriko Iefuji for their lively talks and outstanding help.

From the bottom of my heart I would like to thank my elder brother architect K. Hasibul Kabir for his refreshing mental supports. Especial thank to Golam Kibria Chayon for his cheerful emails and talks. I would like to express my appreciation to my parents, parents-in-law and my sweet sister. I consider myself extremely lucky for the family I have, and at this point I just want to say that it would be never possible for me to come here without their endless support and love.

Last, but not the least, I heartily thank my lovely wife, Shamim Sultana Kakon, for her emotional support and divine sacrifice throughout my study. I would not accomplish this research without her patient understanding and delightful encouragement.

Finally, thanks almighty Allah for giving me constant strength to finish this work.

Contents

List of Figures

List of Tables

CHAPTER 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) [11, 15], e.g., deep space, battle fields, disaster areas, underwater fields, rural areas without infrastructure, sensor networking of distant regions, etc. DTNs cause data communications with long delay, asymmetric data rates, and long queueing delay due to lack of continuous end-to-end connectivity. This class of challenged networks does not adequately well match with the current end-to-end TCP/IP model. In this chapter, a brief background and survey of DTNs, and motivation of the research are discussed.

1.1 Delay Tolerant Networks

The Internet and TCP/IP [28, 54, 58]

To interconnect the communication devices around the globe the Internet becomes a great success. It provides the inter connecting services by using a homogeneous set of communication protocols, called the TCP/IP (Transmission Control Protocol/Internet Protocol) protocol suite. All devices that construct the Internet use these protocols for routing the data and insuring the reliability of message exchanges. Wired-links connection primarily ensures the connectivity on the Internet which includes the wired telephone network. Currently, however, many new wireless technologies, e.g., short-range mobile, satellite links, etc, are also appearing in the Internet technology. To guarantee the connectivity these links should be continuously connected in end-to-end and there should be low-delay paths between sources

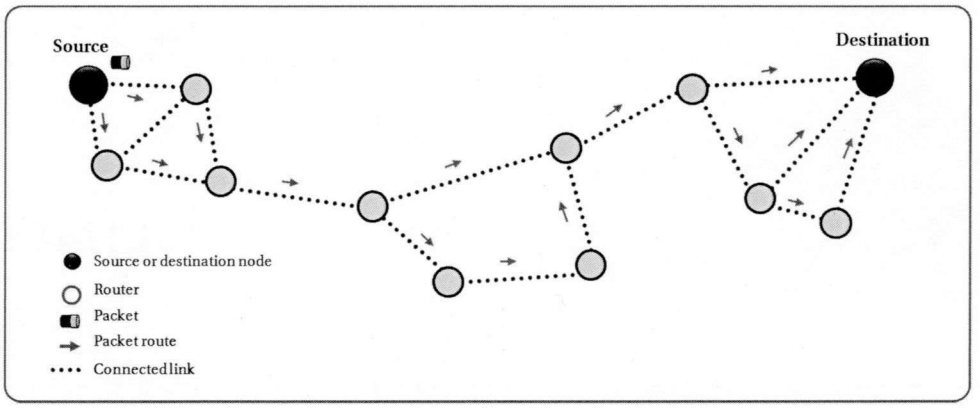


Figure 1.1: Data communication in the Internet.

and destinations as shown in Figure 1.1. The inter connectivity of Internet relies on following important assumptions:

i *Bidirectional and Continuous End-to-End Path*: To establish the end-to-end connectivity a bidirectional continuous connection between source and destination is always required.

ii *Short Round-Trips*: It is required that data round trip is relatively short, i.e., a relatively consistent network delay in sending data packets and receiving the corresponding acknowledgement packets is required.

iii *Symmetric Data Rates*: It is required that in both directions between source and destination a relatively consistent symmetric data rates [1] are provided.

iv *Low Error Rates*: It is required that each link confirms low data loss or relatively low failure of data.

On the the Internet communication is primarily based on packet switching and transfer of data is mainly achieved by protocol layers.

Packet Switching [28, 54, 58] In data communication, packets are defined as pieces of a complete block of user data. Packets travel independently from source to destination through a network of links connected by routers. Each packet that makes up a complete message can take a different path through the network. If one link is disconnected, packets take another link. The header block of a packet contains

[1] Note here that symmetry implies the case where the data rates are symmetric in orders of magnitude.

the destination address and other information that determines how the packet is switched from one router to another [28, 54, 58].

Protocol Layers [28, 54, 58] Messages are moved through the Internet by protocol layers: A set of predefined functions performed by each network node (i.e., network connection point) for data communication between nodes [28, 54, 58]. Host nodes (computers or other communicating devices that are the sources or destinations of messages) usually implement at least five protocol layers, which perform the following functions:

a) Application Layer: The network applications and their application-layer protocol are located at application layer. An application layer protocol is distributed over multiple end systems. The application in one end system uses the protocol to exchange messages with the application in another end system. The application layer of the Internet includes various protocols, e.g., HTTP, SMTP, FTP, etc.

b) Transport Layer: The transport layer is responsible for transporting application-layer messages between the application endpoints. In the Internet, two transport protocols, TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are used. TCP offers connection-oriented service which provides guaranteed delivery of application-layer messages to the destination and confirms data flow control (i.e., speed synchronization between sender and receiver). TCP also segments long messages into shorter segments and provides a congestion control mechanism. On contrary, the UDP protocol provides a connectionless service where there is no reliability, no flow control, and no congestion control.

c) Network Layer/Internet Protocol Layer: The network layer is responsible for routing datagrams, i.e., network layer basic packets of data from one host to another by Internet Protocol (IP). Here, IP provides the basic task of transferring datagrams from source to the final destination with a hierarchical addressing system, i.e., IP addressing. All Internet components that have a network layer must run the IP protocol. The network layer of the Internet also contains routing protocols that determine the routes of the datagrams between sources and destinations.

d) Data Link Layer: To transfer a datagram from a source to its final destination the network layer of the Internet provides the route through a series of packet switches (i.e., routers). The overall data transfer relies on data link layer which provides a raw transmission facility. The link layer packets of data is known as data frames and data link layer sends data frames sequentially. When a packet from one node (host or router) is transferred to the next node in the route, the network layer of host first transfers datagrams down to the link layer which delivers the datagram to the next

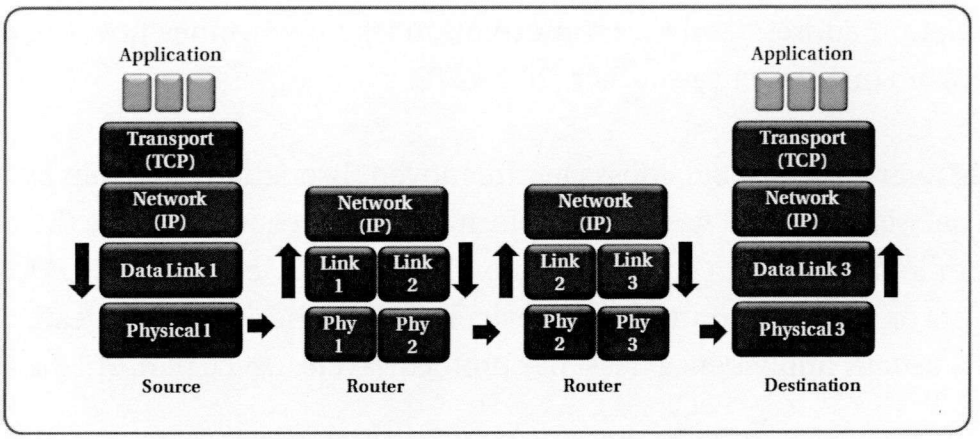


Figure 1.2: Protocol layers in the Internet.

node. From there, the link layer transfer the datagram up to the network layer. The data link layer services depend on the specific link-layer protocol that is employed over the link. As datagrams typically need to traverse several links to travel from source to destination, a datagram may be handled by different link-layer protocols at different links along its route. For a reliable data transfer, the receiver sends an acknowledgment frame to the sender.

e) Physical Layer: The physical layer provides the service to transmit the individual raw bits within the frame from one node to the next. The protocols in this layer are again link dependent, and further depend on the actual transmission medium of the link (in each case, a bit is transmitted across the link in a different way).

In the Internet data communication, routers are used to forward data from one node to another. The Figure 1.2 shows the basic mechanism of protocol layers. Each hop on a path can use a different link-layer and physical-layer technology, and the TCP protocol runs only on source and destination end points, but the IP protocol runs on all nodes. To provide routing path discovery, path selection, name resolution, and error recovery services other Internet protocols and applications are also used [28, 54, 58].

Wireless Network Communications Outside of the Internet [17, 58]

Communications outside of the Internet are mainly defined by the communications in wireless networks, e.g., power limited mobile wireless, sensor nodes, satellite, and interplanetary communications. They are called challenged networks and are accomplished on independent networks, each supporting specialized communication requirements. Figure 1.3 shows an example of the challenged networks. These networks do not use Internet protocols and they are mutually incompatible: each wireless

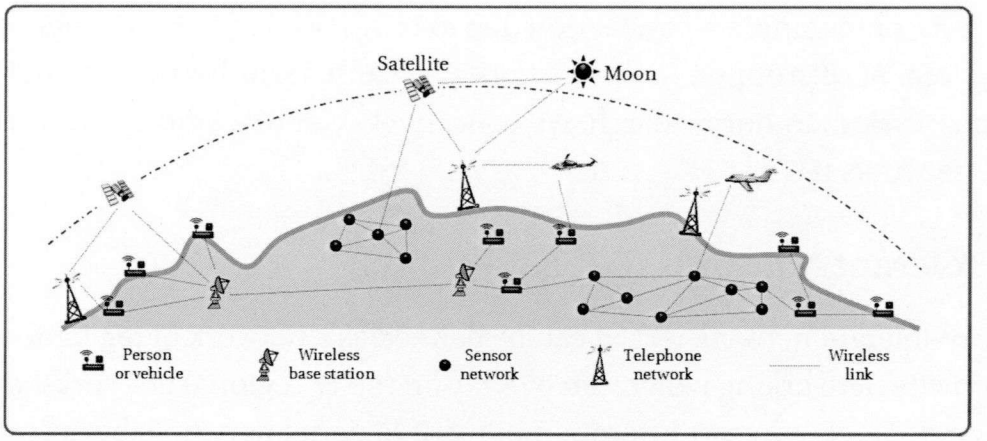


Figure 1.3: Example of the challenged networks.

network is good at passing messages within its network, but not able to exchange messages between networks. Each network is adapted to a particular communication region, in which communication characteristics are relatively homogeneous. The boundaries between regions are defined by link delay, link connectivity, data-rate asymmetry, error rates, addressing and reliability mechanisms, quality-of-service provisions, and trust boundaries. Unlike the Internet, these wireless networks support long and variable delays, arbitrarily long periods of link disconnections, high error rates, and large bidirectional data-rate asymmetries. Examples of wireless networks outside of the Internet include [17, 58]:

- Terrestrial civilian networks connecting mobile wireless devices, including personal communicators, intelligent highways, and remote Earth outposts.
- Wireless military battlefield networks connecting troops, aircraft, satellites, and sensors (on land or in water).
- Outer-space networks, such as the InterPlaNetary (IPN) Internet project [3].

In such networks, an agent is required which can translate between the incompatible networks and can act as a buffer for mismatched network delays.

These kind of challenged networks representing the wireless network communications outside of the Internet are compatible with a new kind of networks called delay tolerant networks. In short, it is DTNs, which is also compatible with conventional Internet. Here, delay implies the end-to-end latency of round trip data transmission which occurs sometimes inherently in the transmission medium, or sometimes due to geometry of the system. Alternatively, the term DTNs is also referred to as disruption tolerant networks, where, disruption implies the factors that cause break down of the

connections, or incomplete connection due to transient or quickly changing aspects of the system. As, disruption in the network also causes long end-to-end delay in the data transmission, in our research, we collectively call this kind of networks delay tolerant networks (DTNs).

Delay Tolerant Networks (DTNs) [1, 17, 58]

A delay-tolerant network (DTN) can be defined as a network of regional networks. DTNs provide networking which can overlay on top of regional networks, including the Internet. DTNs support interoperability of regional networks by accommodating long delays between and within regional networks, and by translating between regional network communication characteristics. In providing these functions, DTNs accommodate the mobility and limited power of evolving wireless communication devices. The wireless DTNs technologies may be diverse, including not only radio frequency (RF) but also ultra-wide band (UWB), free-space optical, and acoustic (sonar or ultrasonic) technologies. Note that the delay tolerant network (DTNs) architecture is originally generated to support the InterPlanetary Internet (IPN). The primary goals of a DTNs are interoperability across network environments, and reliability which can capable of surviving hardware (network) and software (protocol) failures. More information about the DTNs architecture is available at: a) The Internet Research Task Force's Delay-Tolerant Networking Research Group (DTNRG) [1], and b) the InterPlaNetary (IPN) Internet Project [3].

The above mentioned challenged and potential networks (Figure 1.3), do not conform to the underlying assumptions of the Internet. These networks have following characteristics [1, 11, 17, 58] (Figure 1.4) :

(a) *Intermittent Connectivity:* If there is no end-to-end path between source and destination called network partitioning, or if there is discontinuous connection, end-to-end communication using the TCP/IP protocols does not work. Other protocols are required.

(b) *Long or Variable Delay:* In addition to intermittent connectivity, long propagation delays between nodes and variable queuing delays at nodes contribute to end-to-end path delays that can defeat Internet protocols and applications that rely on quick return of acknowledgments or data.

(c) *Asymmetric Data Rates:* The Internet supports moderate asymmetries of bidirectional data rate. But if asymmetries of data rates in order of magnitude are large, they defeat conversational protocols.

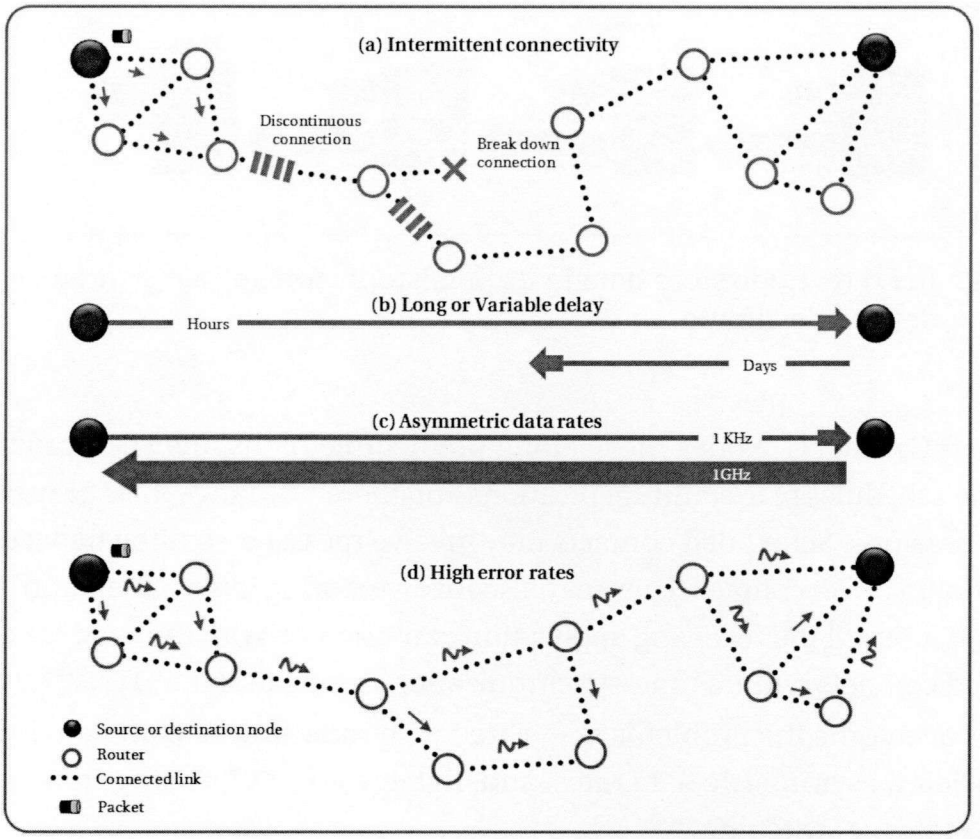


Figure 1.4: Characteristics of DTNs.

(d) *High Error Rates:* Bit errors on links require correction (which requires more bits and more processing) or retransmission of the entire packet (which results in more network traffic). For a given link error rate, fewer retransmissions are needed for hop-by-hop than for end-to-end retransmission (linear increase vs. exponential increase, per hop).

Characterized by DTNs in wireless networks, communications are mainly established by a) opportunistic contacts, and b) scheduled contacts.

Opportunistic Contacts Network nodes may need to communicate during opportunistic contacts, in which a sender and receiver make contact at an unscheduled time. Moving people, vehicles, aircraft, or satellites may make contact and exchange information when they happen to be within line-of-sight and close enough to communicate using their available (often limited) power [1, 11, 17, 58]. For example, wireless Personal Digital Assistants (PDAs) can be designed and programmed to send or receive information when certain people carrying the PDAs come within communication range, or when a PDA is carried past a certain type of information kiosk.

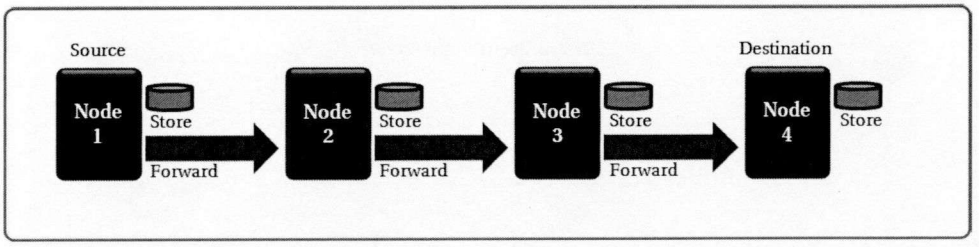


Figure 1.5: In DTNs node store data in its persistent storage, carry and forward it to desired destination.

Scheduled Contacts Nodes move along predictable paths, they can predict or receive time schedules of their future positions and thereby arrange their future communication sessions. Scheduled contacts may involve message-sending between nodes that are not in direct contact. They may also involve storing information until it can be forwarded, or until the receiving application can catch up with the sender's data rate. Scheduled contacts require time-synchronization throughout the DTNs [1, 11, 17, 58].

DTNs overcome the problems associated with intermittent connectivity, long or variable delay, asymmetric data rates, and high error rates by using *store-carry and forward* message switching.

Store-Carry and Forward Message Switching [1, 11, 17, 58]

With store-carry and forward message switching mechanism whole messages (entire blocks of application- program user data) or pieces (fragments) of such messages are moved (forwarded) from a storage place on one node (switch intersection) to a storage place on another node, along a path that eventually reaches the destination, as shown in Figure 1.5.

The storage places (such as hard disk) can hold messages indefinitely. They are called persistent storage, as opposed to very short-term storage provided by memory chips. Internet routers use memory chips to store (queue) incoming packets for a few milliseconds while they are waiting for their next-hop routing table lookup and an available outgoing router port. DTNs routers need persistent storage for their queues for one or more of the following reasons: a) A communication link to the next hop may not be available for a long time, b) one node in a communicating pair may send or receive data much faster or more reliably than the other node, and c) a message, once transmitted, may need to be retransmitted if an error occurs at an upstream (toward the destination) node or link, or if an upstream node declines acceptance of a forwarded message.

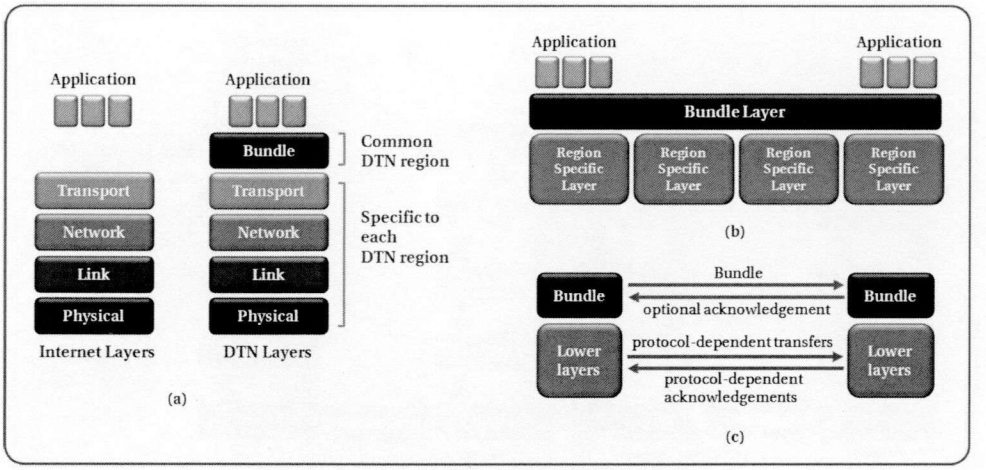


Figure 1.6: Bundle layer in DTNs.

By moving whole messages (or fragments thereof) in a single transfer, the message-switching technique provides network nodes with immediate knowledge of the size of messages, and therefore the requirements for intermediate storage space and retransmission bandwidth.

The DTNs architecture implements store-carry and forward message switching by overlaying a new protocol layer called the *bundle layer* on top of heterogeneous region-specific lower layers.

The Bundle Layer and Bundle Protocol [1, 11, 17, 47, 58]

The bundle layer ties together the region specific lower layers so that application programs can communicate across multiple regions.

Bundle is regarded as the protocol data unit in DTNs which is also called message (as in message-switched). The bundle layer stores and forwards entire bundles (or bundle fragments) between nodes. A single bundle-layer protocol is used across all networks (regions) that make up a DTNs. By contrast, the layers below the bundle layer (the transport layer and below) are chosen for their appropriateness to the communication environment of each region. The Figure 1.6 illustrates the bundle layer as a overlay (top) layer and compares Internet protocol layers with DTNs protocol layers (bottom).

On intermittently connected links with long delays, conversational protocols such at TCP that involve many end-to-end round-trips may take impractical amounts of time or fail completely. For this reason, DTNs bundle layers communicate between themselves using simple sessions with minimal or no round-trips. Any acknowledgement from the receiving node is optional, depending on the class of service

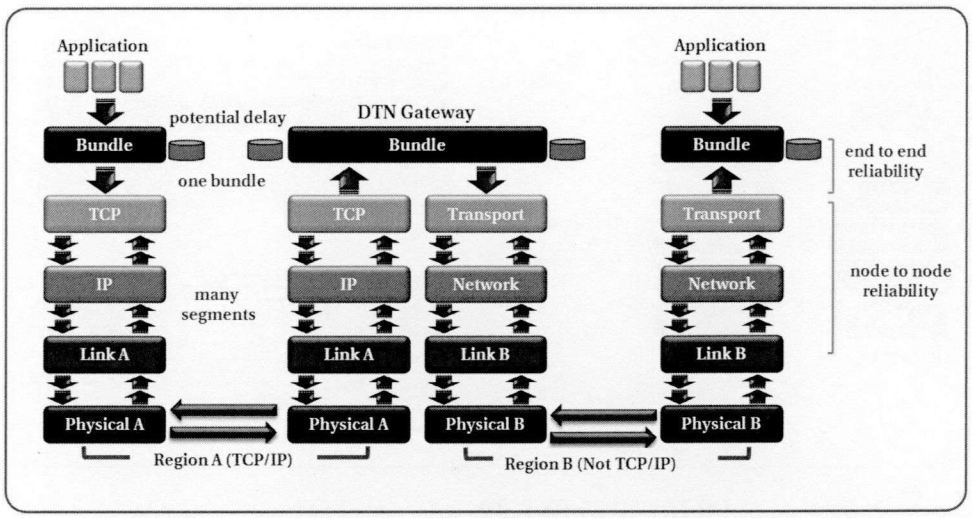


Figure 1.7: Bundle protocol.

selected. The lower-layer protocols that support bundle-layer exchanges may, of course, be conversational like TCP. But on intermittently connected links with long delays, non-conversational or minimally-conversational lower-layer protocols can be implemented (Figure 1.7).

Bundle Protocol: Delay Isolation via Transport Layer Termination [1, 11, 17, 47, 58] On the Internet, the TCP protocol provides end-to-end (source-to-destination) reliability by retransmitting any segment that is not acknowledged by the destination. The network, link, and physical layers provide other types of data-integrity services. In DTNs, the bundle layer relies on these lower-layer protocols to insure the reliability of communication.

However, DTNs routers and gateways nodes that can forward bundles within or between DTNs regions, respectively terminate transport protocols at the bundle layer. The bundle layers thus act as surrogates for end-to-end sources and destinations, as shown in Figure 1.8. The side-effect is that conversational lower layer protocols of low-delay regions are isolated at the bundle layer from long delays in other regions of the end-to-end path. The bundle layer alone supports end-to-end messaging. Bundles are typically delivered atomically, from one node to the next, independent of other bundles except for optional responses, although a bundle layer may break a single bundle into multiple bundle fragments.

On the Internet, the TCP and IP protocols are used throughout the network. TCP operates at the end points of a path, where it manages reliable end-to-end delivery of message segments.

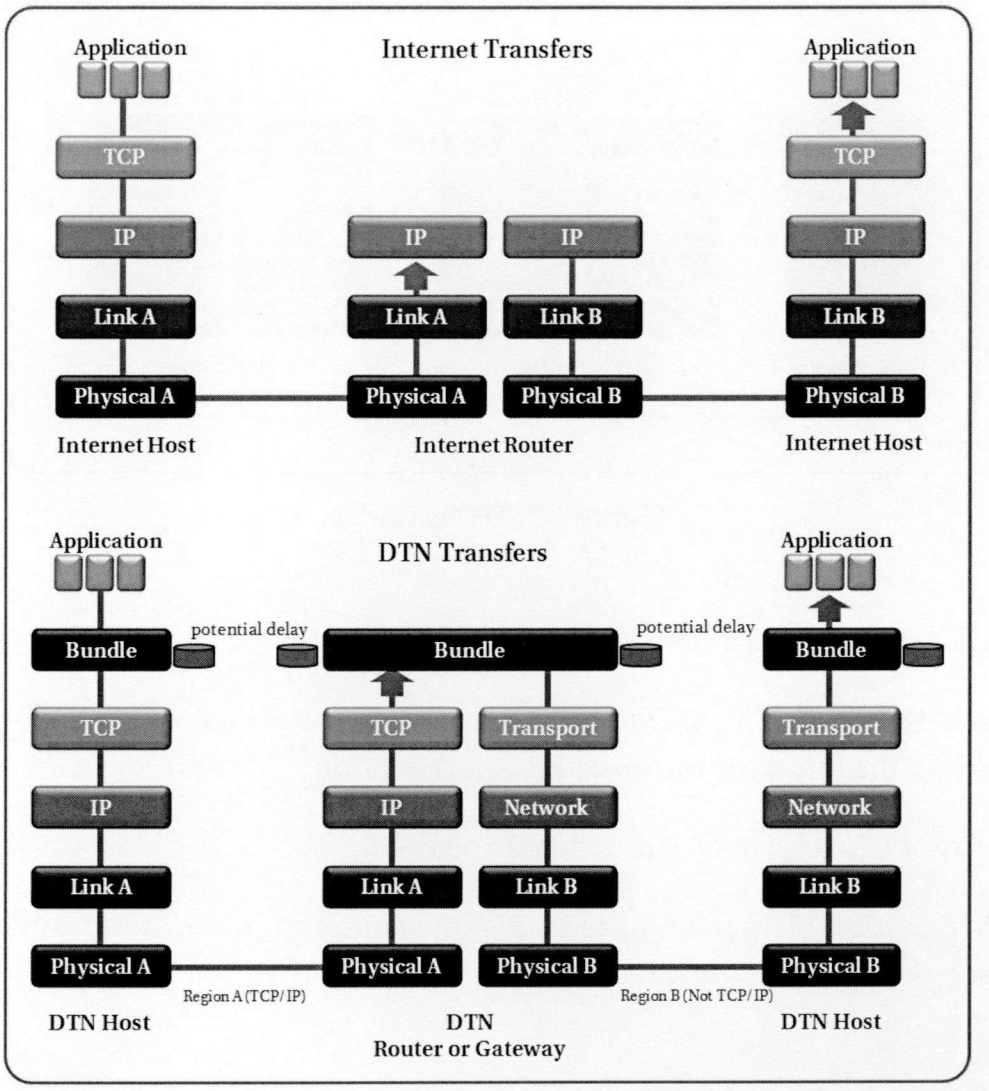


Figure 1.8: Protocol layers and data transfer mechanism in the Internet and in DTNs.

Classes of Bundle Service [1, 11, 17, 47, 58]

The bundle layer provides six classes of service (CoS) for a bundle:

i *Custody Transfer:* Delegation of retransmission responsibility to an accepting node, so that the sending node can recover its retransmission resources. The accepting node returns a custodial-acceptance acknowledgement to the previous custodian.

ii *Return Receipt:* Confirmation to the source, or its reply-to entity, that the bundle has been received by the destination application.

iii *Custody-Transfer Notification:* Notification to the source, or its reply-to entity,

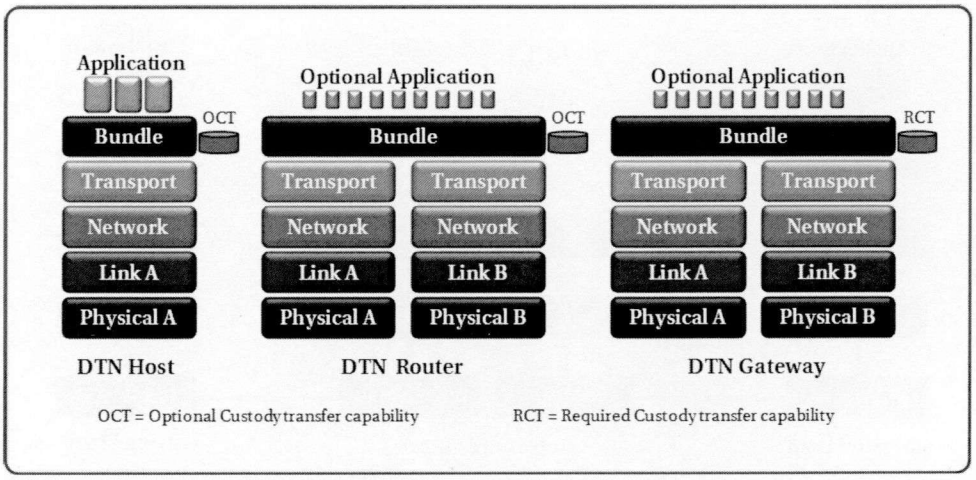


Figure 1.9: DTNs nodes.

when a node accepts a custody transfer of the bundle.

iv *Bundle-Forwarding Notification:* Notification to the source, or its reply-to entity, whenever the bundle is forwarded to another node.

v *Priority of Delivery:* Bulk, Normal, or Expedited.

vi *Authentication:* The method (e.g., digital signature), if any, used to verify the sender's identity and the integrity of the message.

DTNs Nodes

In a DTN, a node is an entity with a bundle layer. A node may be a host, router, or gateway (or some combination) acting as a source, destination, or forwarder of bundles, as shown in Figure 1.9:

- *Host:* Sends and/or receives bundles, but does not forward them. A host can be a source or destination of a bundle transfer. The bundle layers of hosts that operate over long-delay links require persistent storage in which to queue bundles until outbound links are available. Hosts may optionally support custody transfers.

- *Router:* Forwards bundles within a single DTNs region and may optionally be a host. The bundle layers of routers that operate over long-delay links require persistent storage in which to queue bundles until outbound links are available. Routers may optionally support custody transfers.

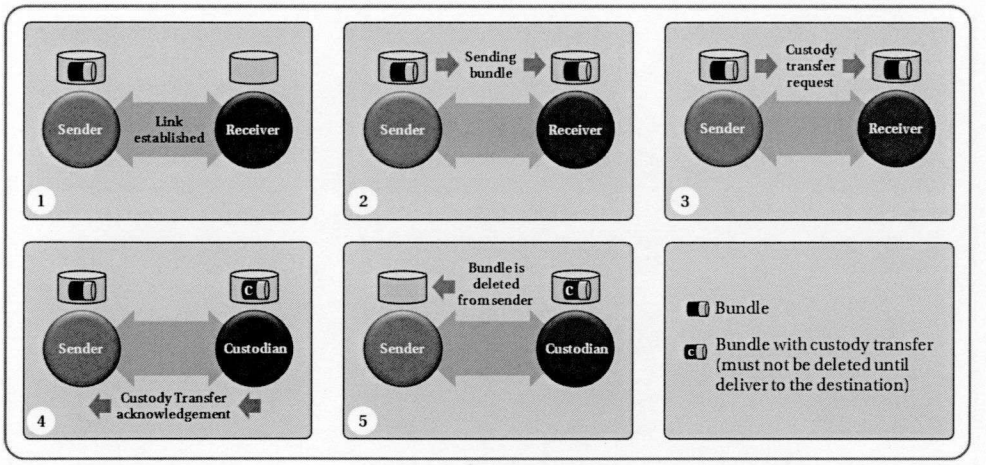


Figure 1.10: Custody transfer mechanism.

- *Gateway:* Forwards bundles between two or more DTNs regions and may optionally be a host. The bundle layers of gateways must have persistent storage and support custody transfers. Gateways provide conversions between the lower-layer protocols of the regions they span.

In a DTN, the protocol stacks of all nodes include both bundle and transport layers. DTNs gateways have the same double-stack layers as DTNs routers, but gateways can run different lower-layer protocols (below the bundle layer) on each side of their double stack. This allows gateways to span two regions that use different lower-layer protocols. For reliable data transfer, DTNs provide *custody transfer mechanism.*

Custody Transfers Mechanism [11, 16, 47]

DTNs support node-to-node retransmission of lost or corrupt data at both the transport layer and the bundle layer. However, because no single transport-layer protocol (the primary means of reliable transfer) operates end-to-end across a DTN, end-to-end reliability can only be implemented at the bundle layer. In what follows, the term retransmission is referred to the bundle layer's retransmission.

The bundle layer supports node-to-node retransmission by means of custody transfers. Such transfers are arranged between the bundle layers of successive nodes, at the initial request of the source application. Nodes with custody transfer request are called *custodians.* As shown in Figure 1.10, a sender first establishes link with a neighboring receiver. Then the sender sends the bundle from its storage to the receiver's storage and starts a time-to-acknowledge retransmission timer. Next, sender requests a custody transfer and if the next-hop bundle layer accepts custody, it returns

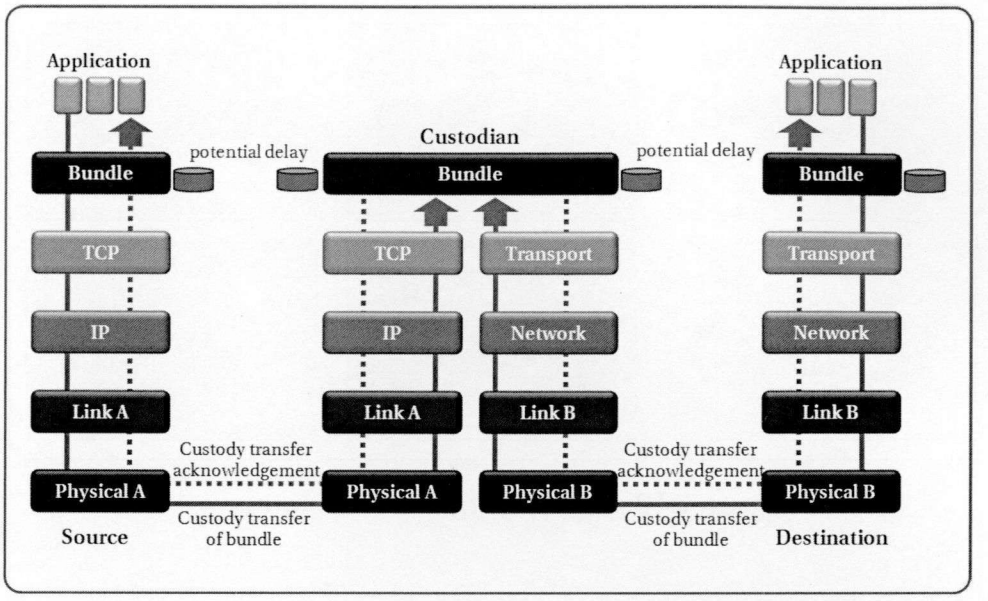


Figure 1.11: Bundle transfer with custody transfer mechanism in DTNs.

an acknowledgment to the sender. If no acknowledgment is returned before the sender's time to acknowledge expires, the sender retransmits the bundle. The value assigned to the time to acknowledge retransmission timer can either be distributed to nodes with routing information or computed locally, based on past experience transmitting to a particular node. After successful transmitting, sender deletes the bundle from its storage to reuse and to prevent from duplicates.

A custodian must store a bundle with custody transfer request until either (1) another node accepts custody, or (2) expiration of the bundle's time to live, which is intended to be much longer than a custodian's time-to-acknowledge. However, the time to acknowledge should be large enough to give the underlying transport protocols every opportunity to complete reliable transmission.

Custody transfers provide guaranteed end-to-end reliability if a source requests both custody transfer and return acknowledgment receipt. In that case, the source must retain a copy of the bundle until receiving a return receipt, and it will retransmit if it does not receive the return receipt.

The bundle layer uses reliable transport-layer protocols together with custody transfers to move points of retransmission progressively forward toward the destination (as shown in Figure 1.11). The advance of retransmission points minimizes the number of potential retransmission hops, the consequent additional network load caused by retransmissions, and the total time to convey a bundle reliably to its destination. This benefits networks with either long delays or very lossy links. For

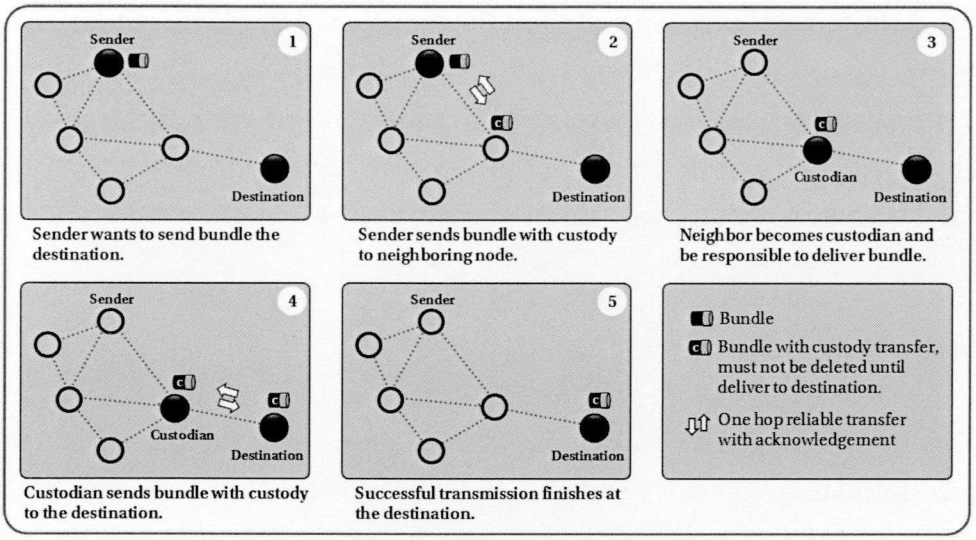


Figure 1.12: Example of reliable bundle transfer in DTNs with store-carry-forward and custody transfer mechanism.

paths containing many lossy links, retransmission requirements are much lower for hop by hop retransmission than for end-to-end retransmission (linear increase vs. exponential increase, with respect to hop count).

In summary, we can conclude the above discussion regarding the characteristics of DTNs that in DTNs, the current TCP/IP model cannot work well due to lack of continuous end-to-end connectivity. Hence, a *store-carry-forward* [11] 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. Therefore, *custody transfer* mechanism [16] is used in store-carry-forward scheme which ensures reliable data transfer among nodes in DTNs: It offers that a bundle with custody must be perfectly delivered from a source to the corresponding destination by delegating the responsibility of reliable transfer with the bundle in a hop-by-hop manner. Figure 1.12 presents a example of reliable bundle transfer in DTNs with store-carry-forward and custody transfer mechanism. 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. Custodians sometimes face storage congestion when they must refuse to receive a new bundle with custody due to lack of their storages or their sufficient energy to keep awake [16]. An increase of bundles with custody and long-term network partitioning accelerate the storage congestion.

In such a situation, some movable vehicles referred to as *message ferries* [63, 65]

can solve the storage congestion problem by actively visiting the network and gather bundles from custodians. If the network is divided into several isolated networks (clusters), message ferries move around the deployment area and deliver bundles among the clusters. In addition, each battery powered node must be awake while holding the bundles to obtain the opportunistic contact of the message ferry.

1.2 Message Ferrying Scheme

There are two message ferry schemes [63,65]: Node-initiated message ferry scheme and ferry-initiated message ferry scheme. In the node-initiated message ferry scheme, ferries move around the deployed area according to known routes and communicate with other nodes they meet. In this scheme, node requires to be mobile. With knowledge of the ferry routes, node that wants to transmit the bundle periodically move close to a ferry route and communicate with the ferry (as shown in Figure 1.13(a)). In the ferry-initiated message ferry scheme, nodes are generally static and the message ferry move proactively to meet nodes. When a node wants to send bundles, it generates a service request and transmits it to the ferry using a long range radio. Upon reception of a service request, the ferry will adjust its trajectory to meet up with the node and collect bundles using short range radios (as shown in Figure 1.13(b)). In both schemes, nodes can communicate with distant nodes that are out of range by using ferries as relays.

In message ferry schemes, most communication involves short range radios. Long range radios are only used in ferry-initiated message ferry for small control messages, avoiding excessive energy consumption. By using ferries as relays, routing is efficient without the energy cost and the network load burden involved in other mobility-assisted schemes that use flooding [63, 65].

In ferry-assisted DTNs, regular nodes are assumed to have assigned tasks and limited in resources such as battery, memory and computation power. Ferries are special mobile nodes which take responsibility for carrying data between regular nodes and have fewer constraints in resources, e.g., equipped with renewable power, large memory and powerful processors. The purposes of ferries are to provide communication capacity between regular nodes.

Message Ferrying is suitable for applications which can tolerate significant transfer delay, such as messaging, file transfer, email, data collection in sensor networks and other non-real-time applications. These applications would benefit from the eventual delivery of data even if the delay is moderate. For example, in a college campus,

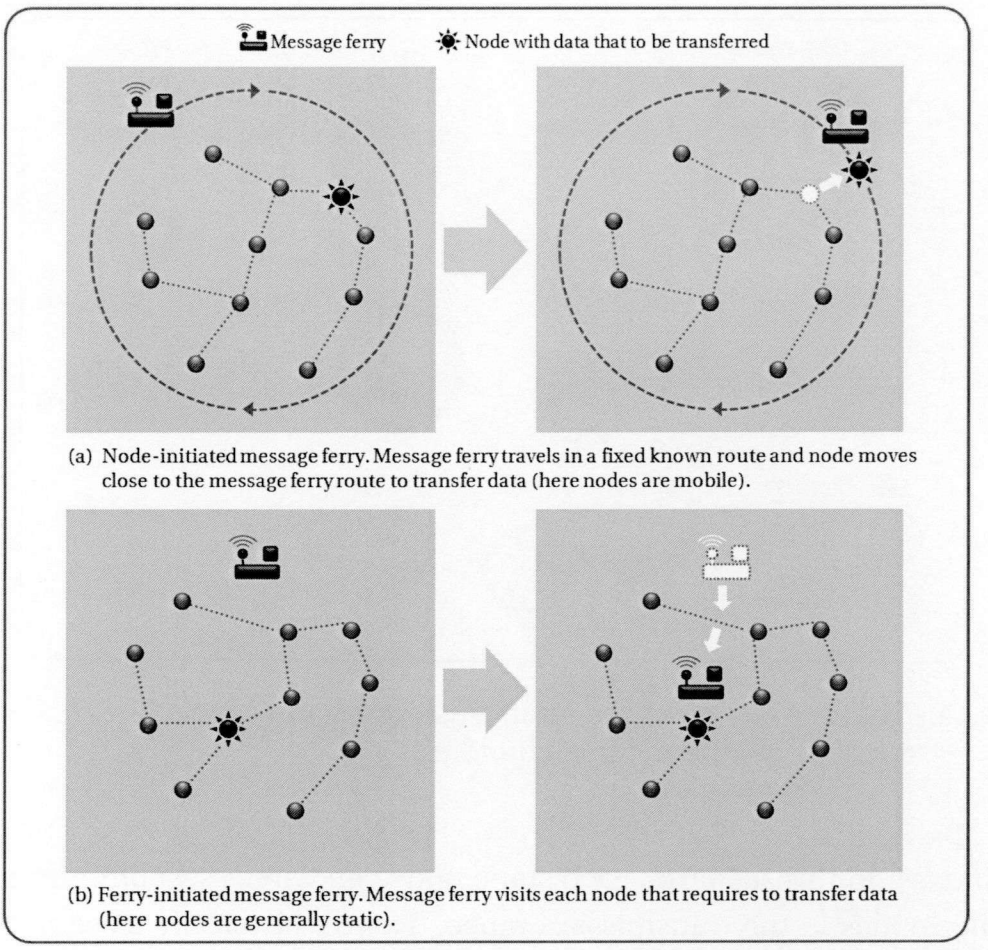


Figure 1.13: Message ferry schemes.

buses equipped with hard disks and wireless interfaces can act as ferries to provide messaging service to students; in battlefield and disaster relief environments, aerial or ground vehicles can be used as ferries to gather and carry data among disconnected areas.

The design of the Message Ferry schemes is based on location awareness and mobility. Each node or ferry is aware of its own location, for example through receiving GPS signals or other localization mechanism.

1.3 Multi-Cluster DTNs

In our proposed DTNs scenario, we aim to achieve a system that periodically collects information from multiple isolated networks, 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 base stations referred to

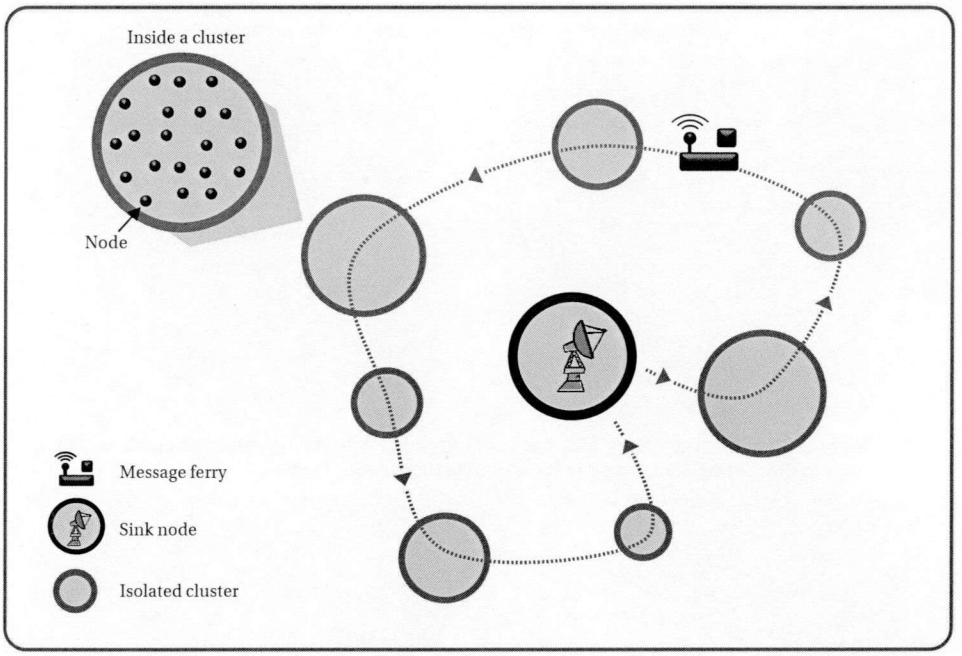


Figure 1.14: Proposed ferry-assisted multi cluster scenario (Size of the cluster proportional to the arrival rate of bundles).

as *sink nodes*. In this scenario, each static node can wirelessly communicate only with other nodes in the transmission range. Hence, physically close nodes form isolated networks referred to as *clusters*. In general, in such DTNs, each node has heterogeneous arrival rate of bundles, and hence, each cluster has average arrival rate of bundles (average heterogeneous offered load). In our proposed DTNs scenario, we consider multiple such kind of clusters which we refer to as *multi-cluster DTNs*. In general, multi-cluster DTNs consists of more than three clusters with several static nodes inside. In such scenario, to collect bundles from the clusters to the sink node, we apply the ferry-initiated message ferry scheme [63], 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 Figure 1.14. We called this scenario as *ferry-assisted multi-cluster DTNs*. This network architecture is suitable for wide area sensing, e.g., DataMULE [48] [2]. Note that in this kind of scenario, message ferries are equipped with a storage enough to carry collected bundles to the destination and it can also supply energy to the custodians if required. The duration of the cycle of the message ferry,

[2]A vehicle (message ferry) that physically carries a wireless compatible communication device between remote locations to efficiently create a DTN data communication link. DataMULE specially offers Internet connectivity to remote villages by attaching wireless compatible computers to buses (message ferries); and while the bus stops at a village, the DTN router on the bus communicates with a DTN router in the bus station over wireless communication.

i.e., duration between message ferry's departs from the sink node and next returns to the sink node, should be as short as possible so that the sink node can grasp the current conditions of all the clusters. When there are so many clusters and/or nodes, the duration tends to be longer. In that situation, we may divide clusters into several groups, based on their locations and the expected amount of generated bundles, and assign a single message ferry to each of those groups. Note that the scheme considered in this research is applicable to such a case because each group of clusters behaves independently.

The duration of the cycle of the message ferry is mainly determined by two factors: The path length of the message ferry and the time for collecting bundles from the clusters and supplying energy to them. In our proposed system, the ferry path/communication is calculated in a hierarchical manner: Inter-cluster communication (the communication between the clusters), and intra-cluster communication (the communication within one cluster, i.e., between the nodes). We assume that the length of the intra-cluster path is negligible compared to that of the inter-cluster path because the distance between nodes in an identical cluster is sufficiently shorter than that between clusters. The sink node can calculate the inter-cluster path in advance by obtaining the information on the physical locations of all clusters.

In such ferry-assisted multi-cluster DTNs scenario, one of the main challenges is to determine a system which can minimize the total mean delivery delay of bundles, where, mean delivery delay defines as the average time interval from the generation of a bundle in a cluster to the completion of its delivery to the sink node. Hence, the objective becomes optimizing inter-cluster communication and intra-cluster communication by taking account of the heterogeneous physical distances of the clusters, heterogeneous arrival rate of bundles where service time of bundles is not negligible, in order to minimize the total mean delivery delay of bundles. Note here that the whole system should be decentralized and autonomous because it is difficult to achieve a centralized control in DTNs due to lack of persistent connectivity.

By taking account of the above objectives, we propose three research studies (Figure 1.15). The inter-cluster communication is focused on by studying the grouping of clusters and the optimal visiting order of isolated clusters. And, the intra-cluster communication is addressed by self-organized data aggregation technique among selfish nodes in an isolated cluster.

Figure 1.15: Proposed research studies.

1.3.1 Grouping Clusters

When there are lots of clusters in ferry-assisted multi-cluster DTNs, multiple message ferries and sink nodes will be required to adopt with the system capacity limit. Hence, clusters can be divided into groups such that each group consists of physically close clusters, a sink node, and a message ferry. In order to minimize the overall mean delivery delay of bundles, group should be created by taking account of the offered load of each cluster, distance between them and capacity limit of each sink node. Hence, a suitable technique is proposed to appropriately locate each base cluster to establish sink node, and to create the groups for the optimal situation.

1.3.2 Visiting order of Clusters

In ferry-assisted multi-cluster DTNs, the message ferry should visit the clusters in a group efficiently to minimize the mean delivery delay of bundles. Hence, the goal is to determine the visiting order of message ferry by taking account of arrival rate, service time, and one-way traveling time between cluster and sink node. A technique

to determine quasi-optimal balanced sequence is proposed in the visiting order of clusters scheme.

1.3.3 Self-Organized Data Aggregation

In a cluster, to collect bundles efficiently from custodian nodes during each visit of the message ferry, bundles can be automatically accumulated to a limited number of nodes (called aggregators). In such case, the message ferry needs to collect the bundles only from the aggregators. This can minimize the traveling distance of the message ferry in a cluster. Self-organized data aggregation technique provides an autonomous and decentralized system to determine the controllable number of aggregators by taking account of the inherent selfishness of the nodes for saving their own battery life.

1.4 Overview of the Thesis

By combining above three research studies, the whole thesis is prepared. A complete system can comprehensively achieve efficient bundle gathering in ferry-assisted multi-cluster DTNs by (i) making groups and determining sink nodes accordingly, (ii) obtaining a visiting order for each group, and (iii) electing aggregators in each cluster.

The thesis is organized in down to top manner according to the above studies.

Chapter 2 addresses intra-cluster communication by self-organized data aggregation technique among selfish nodes in an isolated cluster. We proposed a self-organized data aggregation technique for collecting data from nodes efficiently, which can automatically accumulate data from nodes in a cluster to a limited number of nodes (called aggregators) in the cluster. 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. This chapter is constructed based on the publications in A-1, B-1 and B-2.

Chapter 3 focuses on one part of inter-cluster communication by studying the optimal visiting order of isolated clusters. When there are lots of distant static clusters, the message ferry should visit them efficiently to minimize the mean delivery delay of bundles. We propose an algorithm for determining the optimal visiting order of isolated static clusters in DTNs. We show that the minimization problem of the overall mean delivery delay in our system is reduced to that of the weighted mean waiting

time in the conventional polling model. We then solve the problem with the help of an existing approach to the polling model and obtain a quasi-optimal balanced sequence representing the visiting order. This chapter is constructed based on the publication in A-2.

Chapter 4 focuses on another part of the inter-cluster communication by studying the grouping of clusters. When there are lots of distant static clusters, multiple message ferries and sink nodes will be required. We aim to make groups each of which consists of physically close clusters, a sink node, and a message ferry. Our main objective is minimizing the overall mean delivery delay of bundles in consideration of both offered load of clusters and distance between clusters and their sink nodes. We first model this problem as a nonlinear integer programming, based on the knowledge obtained in our previous work. Because it might be hard to solve this problem directly, we take two-step optimization approach based on linear integer programming, which yields an approximate solution of the problem. This chapter is constructed based on the publications in B-3 and C-1.

Chapter 5 presents the conclusions of this thesis by summarizing all results and observations we obtained through the researches.

CHAPTER 2

Self-Organized Data Aggregation Technique

THIS chapter discusses the intra-cluster communication. Recall that in our proposed system, a fixed sink node collects bundles from nodes in isolated clusters with the help of the message ferry, as shown in Figure 2.1. Each node in a cluster can communicate with other cluster members within the transmission range, called neighbors, but cannot communicate directly with the sink node and/or nodes in other clusters due to the long distances among clusters. The message ferry serves the inter-cluster communication by visiting custodians in each cluster.

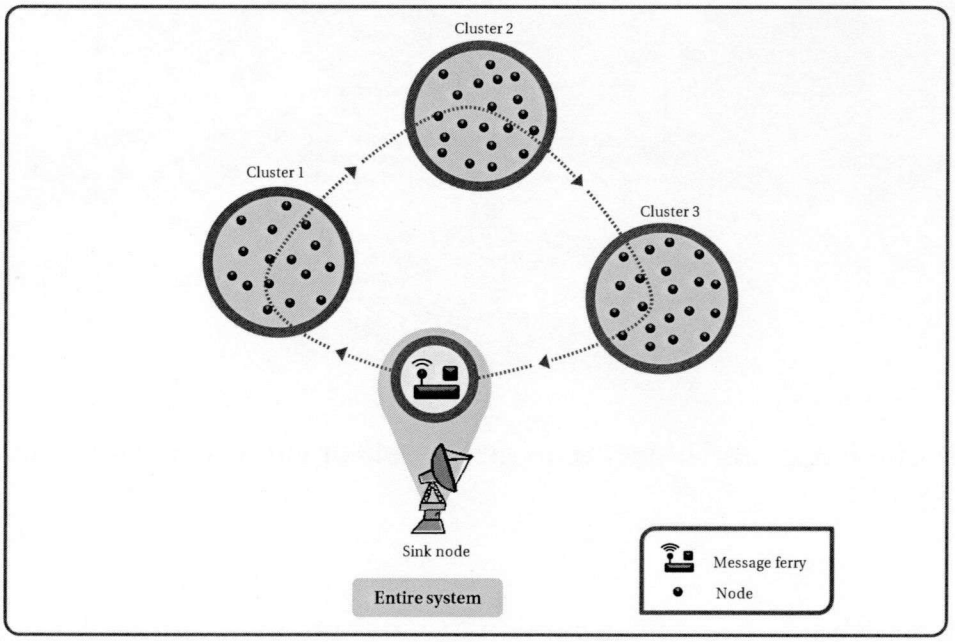


Figure 2.1: Proposed scenario.

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

In long term partitioned networks of our proposed scenarios, 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 may be selfish and reject requests for custody transfer from other nodes to save its storage as well as its energy. This means that the custody transfer mechanism [16] fails without taking the selfishness of custodians into account.

In summary, we face two challenges: a) It is very difficult for message ferries to communicate all storage-congested nodes in a given period of time and 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) take the nodes' selfishness into account (Figure 2.2).

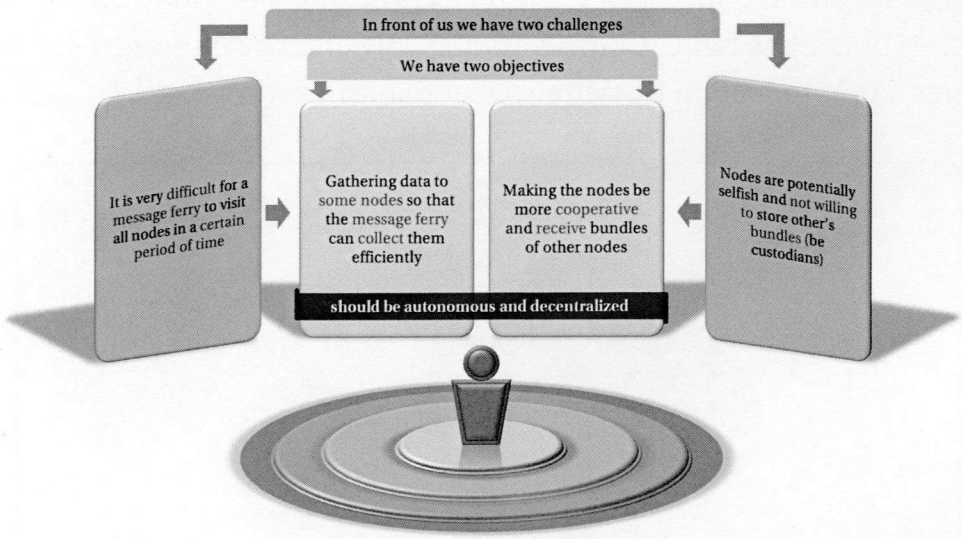


Figure 2.2: Challenges and objective to design self-organized data aggregation technique.

To accomplish such a system, evolutionary game theoretic approach becomes one of the most appropriate mechanisms.

Proposed Method: ***Evolutionary Game Theory:*** Evolutionary game theory origi-

nally explores the dynamics of a population of players under the influence of natural selection [44, 59]. In evolutionary game theory, we assume that fitness (payoff) of a species is determined by not only its own behavior (strategy), which is programmed by genes, but also the behavior of surrounding individuals: the more the fitness is acquired, the larger the population of the corresponding species is [49]. With the help of this scheme, we can finally select some special custodians referred to as *aggregators*, which are cooperative in nature and willingly hold bundles with custody of other nodes.

We developed a self-organized data aggregation technique in [23] by taking account of the challenges. With the help of the evolutionary game theoretic approach [35–37], our system can automatically select some *aggregators*, which are cooperative in custody transfer mechanism with 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 should keep awake to receive bundles from senders anytime and hold bundles until transferring them to the message ferry, while each sender wakes up only when generating and sending bundles, as well as deciding its next role. In addition, each aggregator can obtain energy supply from a message ferry only when it finds a sender among its neighboring nodes. In our scheme, each node appropriately selects its strategy (i.e., being a sender or aggregator), depending on strategies of neighboring nodes. This interaction among nodes is modeled as a game in game theory. The detail will be given in section 2.3.

We examine the characteristics of the proposed scheme by introducing two game models by taking account of the bundle layer's retransmissions mechanism, i.e., reliable transmission of bundles by custody transfer mechanism. Bundle retransmissions are required when a sender cannot find an aggregator in its neighboring nodes. We also introduce a mechanism to adapt to sudden failures of neighboring nodes caused by mismatching of waking time of sender and receiver nodes.

We discuss the system stability through the analysis based on a replicator equation on graphs. Since the replicator equation on graphs focuses only on the strategy distribution, we also investigate the node-level behavior using agent-based dynamics that is a simulation-based approach. Through simulation experiments, we confirm the validity of the analytical results and evaluate the system performance in terms of successful bundle transfer, optimality of aggregator selection, and resilience to node failures.

The rest of the chapter is organized as follows. Section 2.2 reviews the related work. Section 2.3 introduces our self-organized data aggregation scheme. In section 2.4, we analyze the system dynamics and derive the stability condition with the help of a

replicator equation on graphs. We also discuss the system survivability in section 2.5. After a brief introduction of agent-based dynamics, we show some simulation results in section 2.6. Finally, section 2.7 concludes this chapter.

2.2 Related Works

Aggregator selection in our scenario is similar to cluster-head selection in clustering schemes. In general, clustering schemes select some nodes as cluster heads, and then form clusters each of which consists of a cluster head and its physically close nodes. Each cluster head has a responsibility to collect data from cluster members and communicate with other cluster heads. Since the cluster heads consume much energy than normal nodes, several energy-efficient clustering schemes have been proposed. Low-energy adaptive clustering hierarchy (LEACH) [12, 20, 21, 61] and hybrid, energy-efficient, and distributed (HEED) clustering approach [62] are well-known schemes for wireless sensor networks.

LEACH aims to achieve balanced energy consumption among nodes by rotating cluster heads round by round. Each node probabilistically serves as a cluster head, based on a predefined fraction of cluster heads in the network and its role (i.e., a cluster head or normal node) in recent rounds. HEED elects cluster heads in proportion to the residual energy of nodes: Nodes with large residual energy tend to become cluster heads. It has been pointed out that HEED improves network lifetime over LEACH.

These existing approaches work well when all nodes are cooperative. This assumption might crumble in some situations, e.g., when nodes operated by different administrators coexist in the system. In such a situation, nodes are potentially selfish and interested in their own benefit (e.g., battery life), rather than the performance of the whole system (e.g., system lifetime). Evolutionary game theory [22, 50, 59] is useful to model such individual selfishness, which was originally devised to reveal the mechanism that superior genes with high fitness for the environment are inherited from ancestors to offspring, through competition among individuals in the evolutionary process of organisms. In the proposed scheme, aggregator 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.

Evolutionary game theory provides us with both theoretical framework and simulation-based framework. The theoretical framework called *replicator dynamics* is a mathematical model, where the ratio of individuals selecting a strategy increases

when the strategy can yield more payoff than the average payoff of the whole system [22,50,59]. The replicator dynamics is applicable when the population composed of the society is relatively large and well mixed. In actual situations, however, the interactions among individuals are restricted: Each individual knows only a small fraction of members in the society.

To overcome this drawback, Ohtsuki et al. proposed replicator dynamics on graphs by introducing the concept of topological structure into replicator dynamics [35–37]. They derived replicator dynamics on graphs for three kinds of strategy-updating rules: Birth-death updating, death-birth updating, and imitation updating. We will apply imitation updating to aggregator selection, taking account of rational behavior of each node: Each node tries to select a strategy expected to lead to larger payoffs (i.e., residual battery) based on the strategies of neighboring nodes. The detail will be given in the next section.

Inter-cluster communication is also required in DTNs. If the network is partitioned for a long time, the storage congestion frequently occurs in custodians. To alleviate the storage congestion, Zhao et al. proposed message ferry schemes which provide nodes with opportunities of communications among clusters [55, 64, 65]. There are two message ferry schemes [63]: Node-initiated message ferry scheme and ferry-initiated message ferry scheme. In the node-initiated message ferry scheme, nodes know the route of the message ferry in advance and move close to the ferry to transfer bundles on demand, where each node requires to be mobile. On the other hand, in the ferry-initiated message ferry scheme, nodes are generally static and the message ferry takes proactive movement to meet the custodian nodes those require to transfer bundles. After receiving the service request from a custodian, the message ferry proceeds to the custodian and collects bundles. The message ferry can also supply energy to the custodian if required.

In our proposed scenario, in order to communicate static nodes in isolated clusters, we consider ferry-initiated message ferry scheme to collect bundles proactively from custodians. Sometimes it is difficult for message ferries to visit all custodians because of route limitations and traveling costs. In such a case, aggregating bundles to some selected nodes results in reducing the points where message ferries should visit.

2.3 Proposed Scheme

2.3.1 Overview

In our proposed system, in a cluster, the path length of the message ferry is negligible but the time for collecting bundles from nodes and supplying energy to them linearly increases with the number of nodes to be visited. To shorten this time, 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, called cluster members, by local interactions among them. Each non-aggregator (sender) sends its bundles to the aggregators so that the message ferry requires to visit only the aggregators as illustrated in Figure 2.3.

In the above scenarios, we assume that each node is equipped with a long range radio and a short range radio. While the message ferry is approaching a cluster, it broadcasts its availability to all members of the cluster. Only aggregators those paired with sender(s) among their neighboring nodes are allowed to transmit service requests to the message ferry by their long range radio. These service request messages contain the information of each aggregator's location and the amount of bundles it wants to transfer. To guide the message ferry, aggregators occasionally transmit location update messages. On reception of each information, the message ferry calculates the intra-cluster path in an ad hoc manner. When the message ferry and one aggregator are close enough, the aggregator transfers bundles by its short range radio to the message ferry. At the same time, it obtains energy supply from the message ferry. In such situation, wireless energy transfer [29] can reduce the overhead and time for energy supply. Note that the range of long range radio transmission of each aggregator may not necessarily cover the whole deployment area due to power constraints. On the other hand, each sender sends its bundles to the aggregators within the transmission range by its short range radio.

At the initial stage, none of cluster members have any bundles, so they act as senders. While some cluster members generate their own initial bundles, they seek for aggregators within the transmission range. If no aggregator is available, the initial bundle's generators become aggregators. Under cluster members' mutual interactions, aggregators in the next round are selected with the help of evolutionary game theory. We describe the selection procedure of a limited number of aggregators in the next sub-section.

We can summarize the above scenario in each cluster as the repetition of the following three phases (as shown in Figure 2.4):

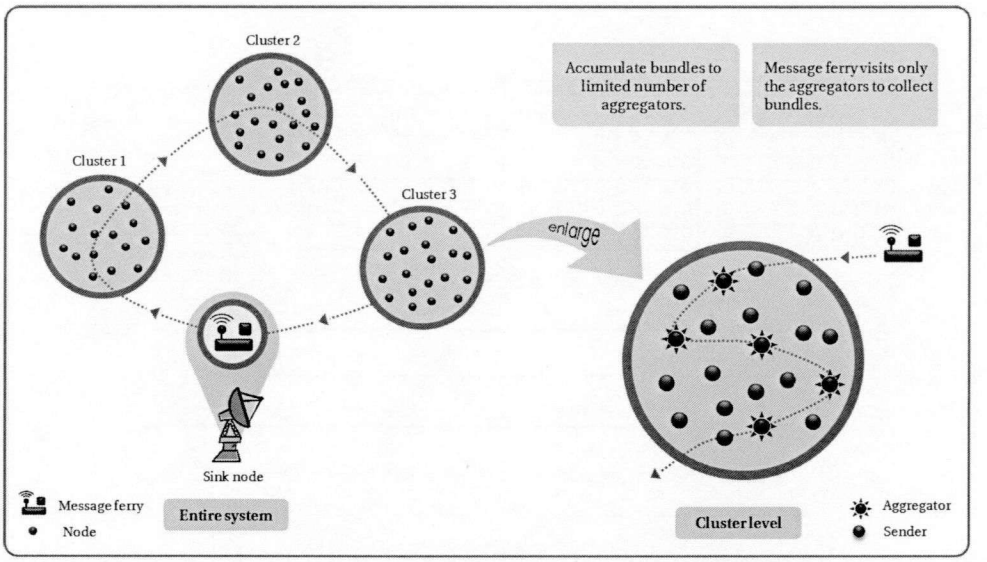


Figure 2.3: Model scenario: Message ferry visits a limited number of aggregators in each cluster and delivers collected bundles to the sink node.

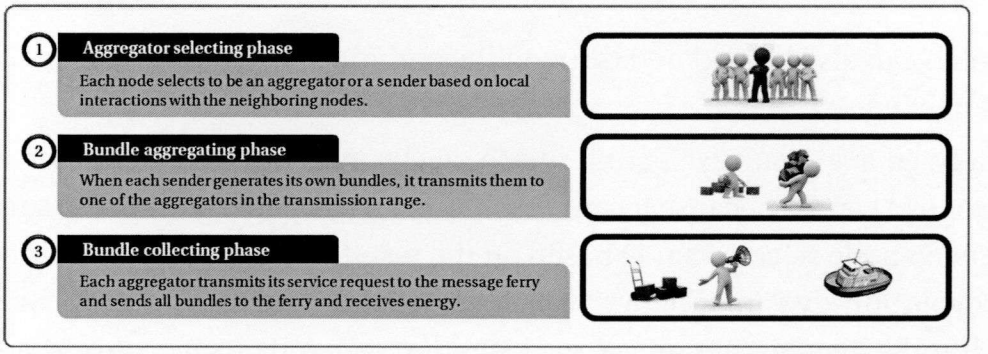


Figure 2.4: Proposed three phase schemes.

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 in the transmission range.

3. *Bundle collecting phase* - Each aggregator transmits its service request to the message ferry and sends all bundles to the ferry. The message ferry supplies energy to aggregators.

We define a *round* as the unit of this repetition. During each round, each node performs these three phases. We presume that all nodes synchronize each other and know the length of the round. The length of the round is pre-determined by the sink

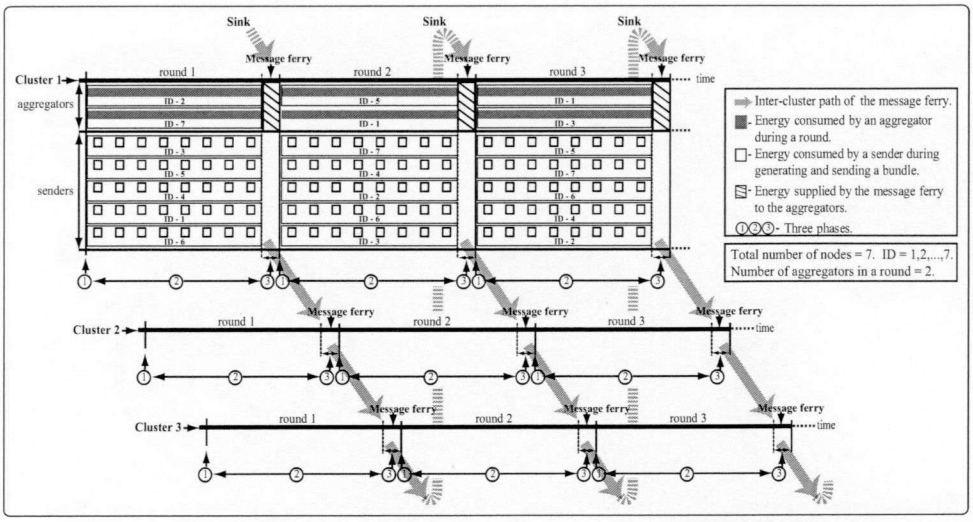


Figure 2.5: Intra-cluster timing chart of cluster 1 with nodes energy consumption (proportional to awaking period) and role of each node (aggregator or sender), and inter-cluster timing chart for clusters 1, 2, and 3.

node which can also be updated through the communication between the ferry and nodes if needed.

Initially, in aggregator selecting phase, each node randomly chooses to be an aggregator or a sender because it cannot know the neighbors' roles. In the subsequent rounds, each node selects its role based on the results of the previous round with the help of evolutionary game theory, whose details are described in later subsections. During bundle aggregating phase, each sender transmits its bundles to one of the aggregators within its transmission range. Then, in the bundle collecting phase, each aggregator transmits its service request to the message ferry, transfers all bundles, and obtains energy supply from the message ferry.

This scenario not only shortens the duration of the round but also gives all nodes benefits in terms of prolonging their battery life. As shown in Figure 2.5, the duration of a round of a cluster depends on the inter-cluster visiting duration. Therefore, during intra-cluster visit in a cluster the message ferry can shorten the duration by visiting only aggregators. On the other hand, in such kind of isolated networks, each custodian node has to consume energy, i.e., battery life by generating its own bundle and awaking all the time by holding bundles until delivering to the message ferry. Note here that the awaking period of nodes is proportional to the energy consumption. There are two ways to keep their batteries in high levels: 1) Obtaining the battery supply from the message ferry at the phase 3 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). In our proposed scenario, aggregators should be awake all the time in the round to receive bundles from senders. As a result, they consume much energy than senders but can also obtain the battery supply from the message ferry. On the other hand, senders cannot obtain the battery supply but can reduce the battery consumption by waking up only when it needs to generate and transmit its own bundle to the aggregators. Figure 2.5 presents the above characteristics of aggregators and senders. We will give more detailed discussion about the battery life in Sect. 2.6.3.6.

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. This can be shown in Figure 2.5: In each round limited number of aggregators is equal to 2 and the role (to be an aggregator or a sender) changes among nodes in each round. We will discuss detail about the role transition of nodes in Sect. 2.6.3.3. These challenges can be divided into two problems: 1) How to select aggregators autonomously under situations where all nodes are potentially selfish, and 2) how to control the number of aggregators. To cope with these problems, we adopt evolutionary game theoretic approach. In the next subsection, we discuss the problem formulation of our proposed scenario based on evolutionary game theory.

2.3.2 Modeling as a game: Selection of the Aggregators

Since it is difficult to achieve a centralized control in DTNs due to lack of persistent connectivity among arbitrary nodes, the selection of aggregators should be realized in a decentralized way. Also, centralized control increases communication overheads among cluster members and it is vulnerable to node failures. More specifically, each node determines to be an aggregator or a sender based on its own benefit, through mutual interaction among neighboring nodes.

We assume that energy consumed by each node in a round increases with the length of time it keeps awake. As illustrated in Figure 2.5 and presented detailed in Figure 2.6, recall that aggregators should always be awake during a round while senders only wake up when generating and transmitting their bundles, as well as deciding their own role. As we already mentioned, all nodes presumed to be synchronized each other and know the length of the round, before the start of each round (just after the message ferry leaves the cluster) all nodes wake up regardless of their current roles and select their next role based on their current conditions (as shown in Figure 2.6).

Let c and s denote the amount of energy consumption per round for aggrega-

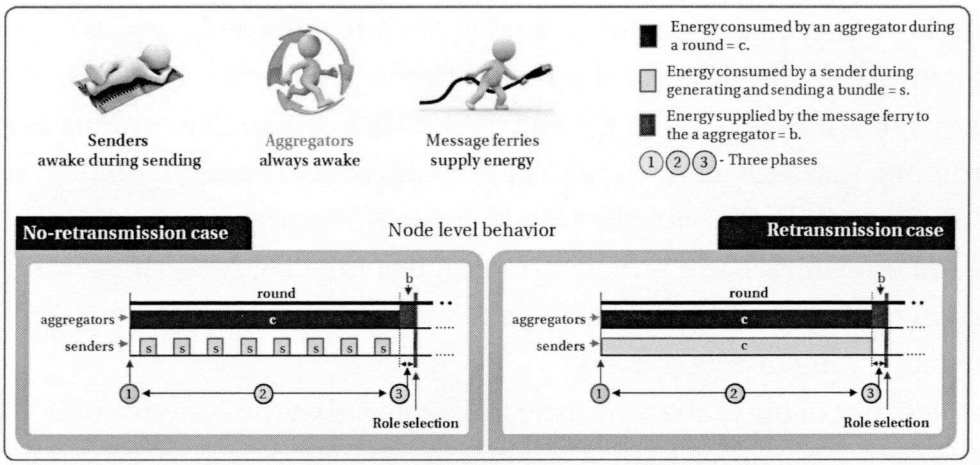


Figure 2.6: Role and node level energy consumption of an aggregator and a sender in one round. During the role selection period all nodes wakeup for a short time and select their next role based on their current conditions.

tors and senders, respectively. s increases with the rate of generating bundles. If retransmissions in the bundle layer do not allow in the sender, we have $c > s > 0$. On the other hand, when senders allow to retransmit bundles without limit, energy consumption of senders increases but never exceeds c. Recall that the bundle layer's retransmissions mechanism, i.e., reliable transmission of bundles by custody transfer mechanism is required when a sender cannot find an aggregator in its neighboring nodes, and failures of transmission in a sender mainly occur due to the mismatch of the waking time of neighboring sender nodes. Next, let b represent the energy supplied by the message ferry to each aggregator. Intuitively, the larger b is, the more the aggregators increase. We assume $b > c$, which is necessary to suppress the number of senders as well as to avoid battery shortages of nodes. Figure 2.6 illustrates the node level behavior for no-retransmission case and retransmission case.

The interaction among nodes can be modeled as a game between two neighboring nodes in evolutionary game theory, which is represented by a payoff matrix. In our scenario, there are two roles (strategies) for each node: An aggregator (aggregate) and a sender (send). Therefore, there are four possible combinations of the strategies of the two nodes, and payoff of each node depends on the combination of strategies. Tables 2.1 and 2.2 illustrate the payoff matrices in the no-retransmission and retransmission cases, respectively. Note that in the payoff matrix, the actions of node 1 form the rows, and the actions of node 2 form the columns. The entries in the matrix are two numerical values representing the payoff of node 1 and node 2 due to the corresponding combination of the strategy, respectively. We discuss detail of them as follows.

Table 2.1: Payoff matrix in no-retransmission case.

node 2 / node 1	sender	aggregator
sender	$-s, -s$	$-s, b-c$
aggregator	$b-c, -s$	$-c, -c$

Table 2.2: Payoff matrix in retransmission case.

node 2 / node 1	sender	aggregator
sender	$-c, -c$	$-s, b-c$
aggregator	$b-c, -s$	$-c, -c$

Table 2.3: Abstract payoff matrix.

node 2 / node 1	sender	aggregator
sender	R, R	S, T
aggregator	T, S	P, P

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 both loose the largest amount of energy $P = c$, because in our proposed system, each aggregator can obtain energy supply from a message ferry only when it finds a sender among its neighboring nodes. On the other hand, an aggregator paired with a sender obtains the largest amount of energy $T = b - c$ because it looses c but obtains b from the message ferry. In this case, the corresponding sender looses the smallest amount of energy $S = s$. When both nodes select to be senders, two possible cases can take place, depending on the presence of bundle layer's retransmissions mechanism. In the no-retransmission case, they both consume $R = s$. On the contrary, in the retransmission case, both of the senders consume $R = c$; this is the worst case that each sender spends all the period of a round on trying to transfer bundle(s) by retransmissions mechanism in bundle layer. Note that we assume that the failure of bundle transfer is mainly caused by the mismatch of waking time of sender and receiver nodes, hence, it needs to retransmit.

2.3.3 Role selection based on imitation and mutation updating

Based on the discussion in the preceding section, we present the abstracted payoff matrix in Table 2.3. We obtained the relationship: $T > S = R > P$ in the no-

retransmission case and $T > S > R = P$ in the retransmission case. In either case, each node not only has a temptation $(T > R)$ to be an aggregator but also a fear $(S > P)$ to be an aggregator. The larger b is, the more the temptation is. This indicates that the sink node can control the number of aggregators by setting b adequately. We show the detail in Section. 2.4. On the other hand, 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 [35]. 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.

The strategy selection in the aggregator selecting phase of the second and subsequent rounds is conduced as follows. At first, each node calculates its own payoffs according to the strategies of its neighbors, as well as its own strategy, in the previous round. It then chooses between keeping its current strategy or imitating one of the neighbor's strategies proportional to payoffs, because it is selfish in nature and aims to maximize its own payoff, i.e., prolonging its own battery life. This strategy updating is called *imitation updating* in [37]. As a result, the replicator equation on graphs derived in [37] is applicable to our system, and we can obtain the relationship between the parameters, i.e., b, c, s, the average number of neighbors, and the fraction of aggregators in the system.

Even though imitation updating ordinarily works well, it has one drawback: Each node cannot change its strategy when all neighbors take the same strategy as that the node takes. Each node wants to avert such a situation because its main purpose is sending bundles to the sink node via the message ferry. To cope with this problem, we also consider a system with *mutation updating* [57]. In [57], mutations occur with probability $P_u \geq 0$ in each round, while in our case, a node changes its strategy with mutation probability P_u only when the node and all the neighbors take the same strategy.

In the next sections, we clarify the relationship between the parameters and the ratio of number of aggregators using evolutionary game theory.

2.4 Analytical Results

In this section, we discuss the relationship between the ratio of aggregators, parameters of the payoff matrix, topological structure, and updating rules. We consider the

Table 2.4: Modifier matrix.

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

proposed scheme without mutation updating (i.e., $P_u = 0$). To reveal the relationship among those, we consider the replicator equation on graphs in evolutionary game theory [37, 38]. 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 our system. We give the details of evolutionary game theory and the replicator equation on graphs in Appendix A.

2.4.1 Replicator Equation on Graphs: System Dynamics and Stability

We first briefly introduce the replicator equation on graphs in [37] and model our proposed system for no-retransmission case to match adequately. There are n strategies labeled i ($i = 1, 2, \ldots, n$) and the payoff of strategy i versus strategy j is denoted by $a_{i,j}$. The relative frequency of strategy i is given by x_i, where $\sum_{i=1}^{n} x_i = 1$. The average payoff of strategy i is then given by

$$f_i = \sum_{j=1}^{n} x_j a_{i,j}. \tag{2.1}$$

In our system, we consider two kinds of strategies, e.g., aggregating and sending. In what follows, we regard strategies 1 and 2 for aggregator and sender, respectively, i.e., x_1 denote the ratio of the number of aggregators to the total number of cluster members and $x_2 = 1 - x_1$ represents the ratio of the number of senders. Following Table 2.1 the expected payoff f_1 and f_2 of aggregators and senders are given by

$$f_1 = (1 - x_1)(b - c) - cx_1, \quad f_2 = -s, \tag{2.2}$$

respectively.

Let k denote the number of neighbors of each node, called degree [36]. Although the equation is derived for k-regular graph for $k \geq 3$, it is also applicable to non-regular

graphs, e.g., unit disk graph, random networks, scale free networks, etc [36, 37]. In such a case, k represents the average degree ($k_{avg} \geq 3$). The modified payoff matrix for evolutionary game theory on graphs is defined as the sum of the original payoff matrix and a modifier matrix [37]. Table 2.4 shows the modifier matrix, where m describes the local competition between the strategies [37]. The gain of one strategy is the loss of another and local competition between the same strategies results in zero. It follows from Eq. (A.5) that m becomes

$$m = \frac{3b - (k+6)(c-s)}{(k+3)(k-2)}, \quad k \geq 3. \tag{2.3}$$

Follows from Eq. (A.6)the expected payoff for the local competition g_1 and g_2 of aggregators and senders are obtained to be

$$g_1 = (1 - x_1)m, \qquad g_2 = -x_1 m, \tag{2.4}$$

respectively, where m is given by Eq. (2.3). The average payoff ϕ of two strategies is then given by

$$\begin{aligned} \phi &= x_1(f_1 + g_1) + (1 - x_1)(f_2 + g_2) \\ &= (1 - x_1)(bx_1 - s) - cx_1. \end{aligned} \tag{2.5}$$

From Eqs. (2.2), (2.4), and (2.5), we obtain the replicator equation on graphs [37] in the no-retransmission case to be

$$\begin{aligned} \dot{x}_1 &= x_1(f_1 + g_1 - \phi) \\ &= x_1(1 - x_1)\left[\frac{b(k^2 + k - 3) - (c-s)(k^2 + 2k)}{(k+3)(k-2)} - bx_1\right], \quad k = 3, 4, \ldots. \end{aligned} \tag{2.6}$$

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

$$x_1^* = 1 - \frac{c-s}{b} + \frac{1}{(k+3)(k-2)}\left[3 - (k+6)\cdot\frac{c-s}{b}\right], \quad k = 3, 4, \ldots. \tag{2.7}$$

Note that the equilibrium in Eq. (2.7) is feasible if

$$0 < x_1^* < 1,$$

or equivalently,

$$\max\left(c, \frac{k(k+2)}{(k+3)(k-2)}\cdot(c-s)\right) < b < \frac{k(k+2)}{3}\cdot(c-s), \quad k = 3, 4, \ldots. \tag{2.8}$$

Because $0 < s < c < b$ and $1 < (k^2 + 2k)/(k^2 + k - 3) \leq 5/3 < 15/3 \leq (k^2 + 2k)/3$ for all $k \geq 3$, there always exists b ($b > c$) that satisfies Eq. (2.8). Thus the equilibrium in Eq. (2.7) is controllable. Furthermore, x_1^* in Eq. (2.7) is stable for such b because $\dot{x}_1 > 0$ if $0 < x_1 < x_1^*$, and otherwise, $\dot{x}_1 < 0$.

Similarly, in the retransmission case, with the help of the payoff matrix in Table 2.2, the stable and controllable equilibrium becomes

$$x_1^* = 1 - \frac{c-s}{b+c-s} + \frac{3}{(k+3)(k-2)}\left[1 - \frac{2(c-s)}{b+c-s}\right], \quad k = 3, 4, \ldots. \tag{2.9}$$

that is valid for

$$c < b < \frac{(k+3)(k-2)}{3} \cdot (c-s), \quad k = 3, 4, \ldots. \tag{2.10}$$

In the same way as for Eq. (2.8), we can show that there always exists b ($b > c$) satisfying Eq. (2.10) and that x_1^* in Eq. (2.9) for such b is stable. In what follows, we call Eqs. (2.8) and (2.10) as *stability conditions.*

Note that in equilibrium, the fraction of aggregators in the system is fixed but the strategy of each node will change round by round [23]. This feature is suitable for our system because each node can acquire opportunities to obtain energy supply when it serves as an aggregator.

2.4.2 Valid parameter settings for permanently living system

Although each node has a chance of obtaining energy supply, careful parameter tuning is required to achieve high system survivability. We assume that in equilibrium, each sender can find an aggregator among neighbors and vice versa. Thus each sender consumes s units of energy and each aggregator obtains $b - c$ units of energy in equilibrium. The expected payoff of each node per round is then given by

$$\mathrm{E}[p] = (b-c)x_1^* - s(1-x_1^*),$$

where, $E[p]$ represents the expected amount of energy supplied to a randomly chosen node per round. Note that $\mathrm{E}[p]$ should be positive because the system would die out if $\mathrm{E}[p] = 0$. In our proposed system, by adequately adjusting the amount of initial energy of each node, the system can handle losses of nodes. Moreover, during low energy level a node can change its role to become an aggregator so that it can be allowed to get energy supply from the message ferry in the next round. We will give more detailed discussion about the battery life in Sect. 2.6.3.6. Therefore, for $\mathrm{E}[p] > 0$,

b, c and s should satisfy the following inequality

$$x_1^* > \frac{s}{b-c+s}. \tag{2.11}$$

In what follows, we call Eq. (2.11) *running condition.*

In practice, the sink node tries to achieve a *feasible* x_1^* that 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 and the generation rate of bundles. The average node degree, k, is also given from the environment. As a result, the sink node achieve desirable and feasible x_1^* by mainly controlling b.

2.4.3 Discussions

It is preferable to achieve small, feasible x_1^*. Therefore we discuss the infimum of feasible x_1^* for fixed k, c, and s. Recall that in no-retransmission case, the equilibria $x_1^*(b) := x_1^*$ is an increasing function of b, while $x_1^*(b)$ should satisfy Eq. (2.8) and (2.11), i.e.,

$$\frac{s}{b-c+s} < x_1^*(b) < 1.$$

As a result, the infimum of feasible $x_1^*(b)$ is obtained for $b = b^*$ such that

$$x_1^*(b^*) = \frac{s}{b^*-c+s}, \tag{2.12}$$

Note here that $s/(b-c+s)$ is a decreasing function of b. Therefore $x_1^*(b) > s/(b-c+s)$ for all b ($b > b^*$). Furthermore, $0 < s/(b-c+s) < 1$, so that $0 < x_1^*(b^*) < 1$.

We first consider the no-retransmission case. It follows from Eqs. (2.7) and (2.12) that

$$(k^2+k-3)(y^*)^2 - (k+3)(c-s+(k-2)s)y^* - (k+3)(k-2)(c-s)s = 0, \tag{2.13}$$

where $y^* = b^* - c + s$. Recall that $x_1^*(b^*) = s/y^*$. Substituting $y^* = s/x$ into Eq. (2.13) and rearranging terms yield

$$f_{\mathrm{No}}(x) = (k+3)(k-2)(c-s)x^2 + (k+3)(c-s+(k-2)s)x - (k^2+k-3)s = 0. \tag{2.14}$$

We then have

$$x^*_{\mathrm{No}} = x^*_1(b^*) = \frac{1}{2}\left[\sqrt{\left(\frac{1}{c/s-1}+\frac{1}{k-2}\right)^2 + 4\left(1+\frac{3}{(k+3)(k-2)}\right)\frac{1}{c/s-1}} - \left(\frac{1}{c/s-1}+\frac{1}{k-2}\right)\right]. \tag{2.15}$$

It is easy to see that $x^*_1(b^*)$ goes to zero as $c/s \to \infty$. Furthermore, it can be shown that $x^*_1(b^*) \leq 1/2$ if $c/s \geq 3 - 4/(k+3)$.

Next we consider the retransmission case. According to the same procedure as in the no-retransmission case, we obtain the following equation corresponding to Eq. (2.14).

$$f_{\mathrm{Re}}(x) = 2(k+3)(k-2)(c-s)x^2 - (k+3)(k-2)(c-2s)x - (k^2+k-3)s = 0, \tag{2.16}$$

and therefore

$$x^*_{\mathrm{Re}} = x^*_1(b^*) = \frac{1}{2} + \frac{1}{4}\left[\sqrt{\left(\frac{c/s}{c/s-1}\right)^2 + 4\left(1+\frac{6}{(k+3)(k-2)}\right)\frac{1}{c/s-1}} - \frac{c/s}{c/s-1}\right]. \tag{2.17}$$

It is easy to see that $x^*_1(b^*) > 1/2$ and it goes to $1/2$ as $c/s \to \infty$.

We now compare x^*_{No} and x^*_{Re}, which are given by the larger real roots of convex, quadratic equations Eq. (2.14) and (2.16), respectively. See Figure 2.7 that plots $f_{\mathrm{No}}(x)$ and $f_{\mathrm{Re}}(x)$ for $k = 3$, $c = 2$, and $s = 1$. It can be shown that

$$f_{\mathrm{No}}(x) - f_{\mathrm{Re}}(x) > 0 \quad \text{for} \quad x > 0 \qquad \Leftrightarrow \qquad 0 < x < 1 + \frac{1}{k-2}.$$

Because $0 < x^*_{\mathrm{Re}} < 1$, we have $f_{\mathrm{No}}(x^*_{\mathrm{Re}}) - f_{\mathrm{Re}}(x^*_{\mathrm{Re}}) > 0$, so that

$$x^*_{\mathrm{No}} < x^*_{\mathrm{Re}}, \tag{2.18}$$

for any fixed k, c, and s.

Owing to Eq. (2.12), Eq. (2.18) implies that b^* in the no-retransmission case is greater than b^* in the retransmission case. As shown in Tables 2.1 and 2.2, the no-retransmission case gives nodes less fear to become senders than the retransmission case. As a result, the no-retransmission case requires larger b to generate aggregators adequately. In the next subsection, we show some numerical results to illustrate the

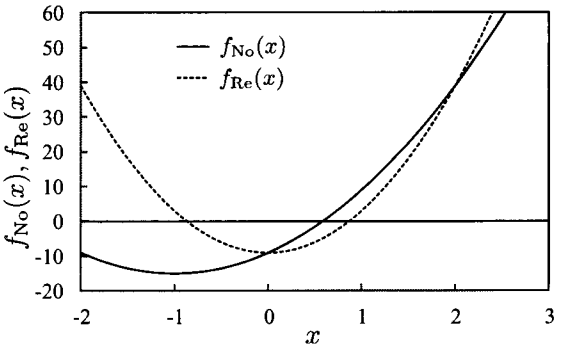


Figure 2.7: Quadratic functions $f_{\mathrm{No}}(x)$ and $f_{\mathrm{Re}}(x)$ ($k = 3$, $c = 2$, $s = 1$).

feasible parameter settings.

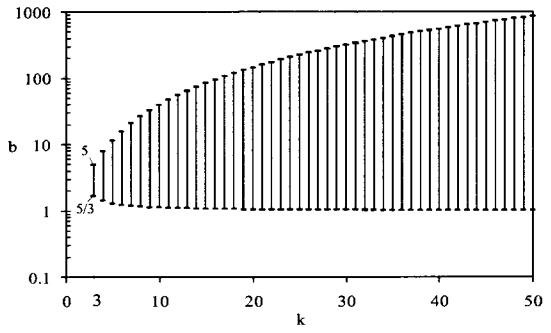
2.5 Numerical Results

In this section, we show some numerical examples of the adequate parameter settings according to the theoretic analysis in sections 2.4.1 and 2.4.2.

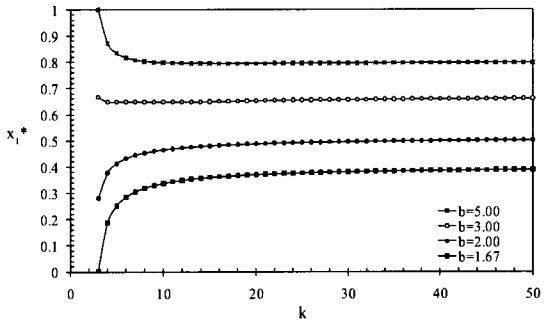
We have four independent variables, b, c, s and k, which affect x_1^*. First we observe the results for no-retransmission case. Here for simplicity, $c - s$ is assumed to be one. Note that this simplification does not lose generality. Note here that the (average) degree k is a pre-determined parameter representing the density of nodes in the system under consideration. As a result, the ratio of aggregators can be controlled only by b according to Eq. (2.7). The expected number of aggregators can be obtained by the product of x_1^* and the number of cluster members.

Figure 2.8(a) illustrates the range of b with the supremum and infimum that satisfy Eq. (2.8), as a function of k. We observe that the valid range of b widens with k, while the infimum is almost constant. Figure 2.8(b) shows the controllable equilibrium x_1^* as a function of k. As shown in Eq. (2.7), x_1^* can take any value between 0 and 1 in both cases, depending on b and k. From those figures, we observe that for each b, x_1^* does not change when k becomes large, because the modifier m converges to zero with an increase of k, as shown in Figure 2.8(c). We also observe that for a fixed k, the small b leads to the small x_1^*, which can be shown analytically with Eq. (2.7). When k is less than 20, the controllable equilibrium x_1^* shows different characteristics, depending on b. Roughly speaking, if the modifier m is negative (i.e., $b < 3$), x_1^* is a non-decreasing function of k, and otherwise, x_1^* is a non-increasing function of k.

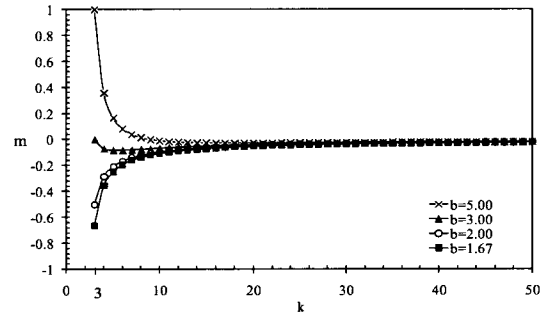
Finally, for a given k, Figure 2.8(d) shows appropriate values of b to achieve a specific value of x_1^*, where x_1^* is set to be 0.1, 0.3, and 0.5. We first find that b can be less



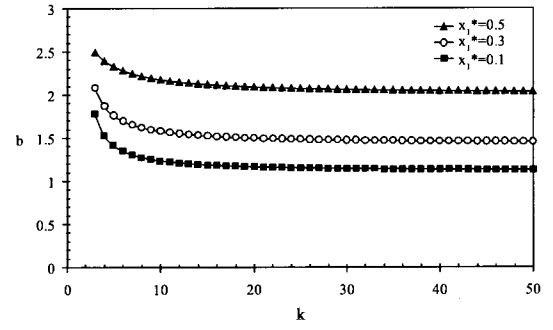
(a) The supremum and infimum of b ($c - s = 1$).



(b) The controllable equilibrium x_1^* ($c - s = 1$).



(c) The modifier m ($c - s = 1$).

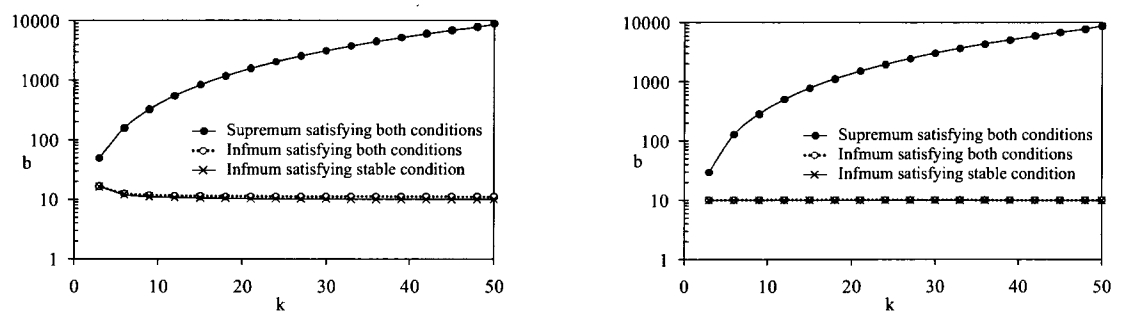


(d) Appropriate b to achieve x_1^* ($c - s = 1$).

Figure 2.8: The supremum and infimum of b satisfying stable and/or running conditions (No-retransmission case).

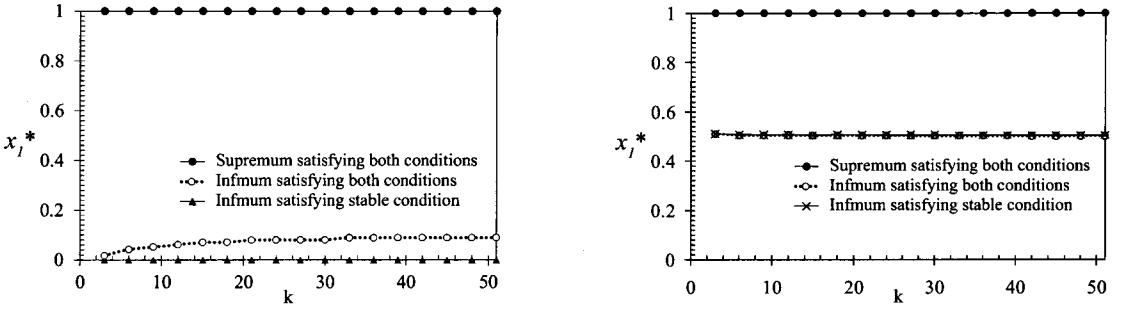
than 3.00. Note that this value of b is valid under the assumption of $c - s = 1$. We do not need much larger b to achieve our objective that is limiting the ratio of aggregators. Furthermore, if k is larger than 20, b converges to a value, depending on the target level of x_1^*.

Next, we clarify the impact of stable and/or running conditions and the effect of retransmission mechanism. Figure 2.9 depicts the valid range of controllable benefit b as a function of k when $c = 10$ and $s = 0.1$. Figure 2.10 illustrates the corresponding range of x_1^*. Note that we 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, Figure 2.10 indicates that the retransmission mechanism narrows the valid range of equilibrium compared with no-retransmission case. Specifically, x_1^* must be greater than 0.505 to satisfy both conditions in retransmission case.



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

Figure 2.9: The supremum and infimum of b satisfying stable and/or running conditions (c=10, s=0.1).

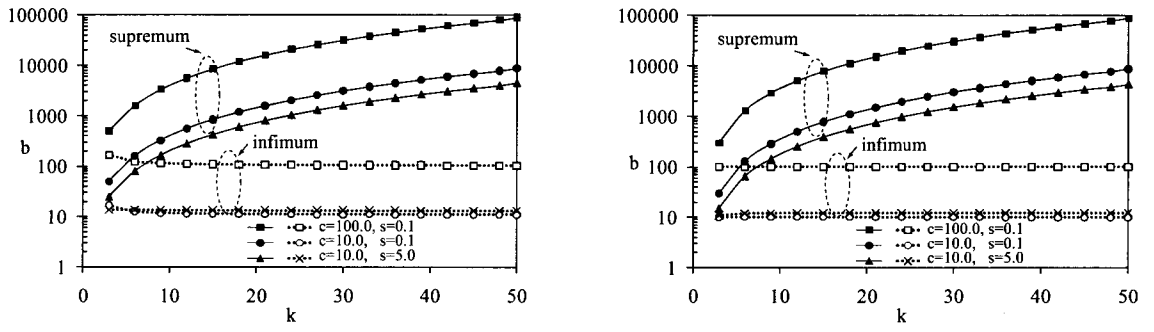


(a) No-retransmission case. (b) Retransmission case.

Figure 2.10: The supremum and infimum of x_1^* satisfying stable and/or running conditions (c=10, s=0.1).

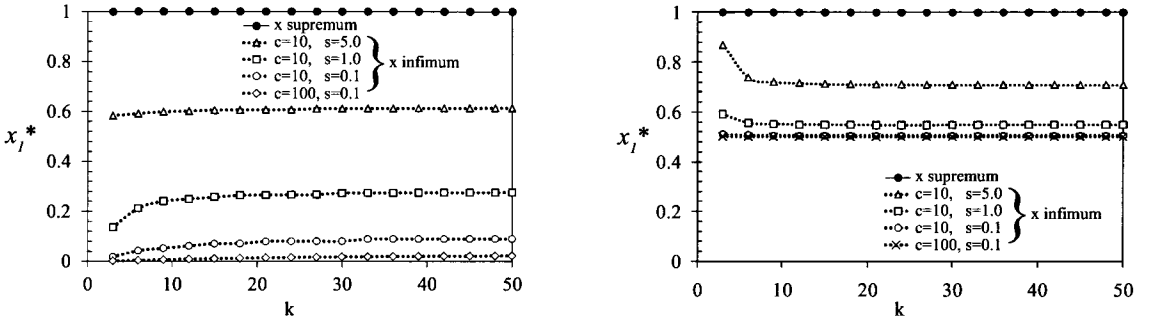
Next, we reveal how c and s affect the valid range of b and x_1^*. Figure 2.11 illustrates the supremum and infimum of b satisfying stable and/or running conditions when c and s vary. We observe that for a specific k, the range of b shifts up with c. This simply means that $b - c$ should be positive. 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.

Figure 2.12 presents the supremum and infimum of x_1^* corresponding to Figure 2.11. We observe that s has more impact on infimum than c. This is mainly caused by running condition. From Eq. (2.11), 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.

Figure 2.11: Effect of c and s on the supremum and infimum of b satisfying stable and/or running conditions.



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

Figure 2.12: Effect of c and s on the supremum and infimum of x_1^* satisfying stable and/or running conditions.

2.6 Simulation Experiments

Replicator dynamics is a powerful mathematical tool to predict the macro-level system behavior and it clarifies the effect of parameters on it. However, we can gain little insight into the micro-level system behavior such as the influence of irregularity of the topology on the system behavior, the geographical distribution of strategies, transient phenomena (including the convergence time to the equilibrium), and so on. Therefore we conduct simulation experiments based on agent-based dynamics, which is a complementary method to understand the micro-level system behavior in the evolutionary game theory. It models such a phenomenon that a superior strategy spreads over the network in a hop-by-hop manner, where local interactions among neighboring nodes are defined explicitly. In agent-based dynamics, each node interacts with neighboring nodes in every round and determines its strategy for the next round based on the acquired payoffs. In the case of imitation updating,

each node chooses between keeping its current strategy and imitating one of the neighbors' strategies proportional to the payoff [37]. In what follows, we conduct simulation experiments of agent-based dynamics for two purposes, the validation of the analytical results, and the investigation of the micro-level system behavior.

2.6.1 Agent-based Dynamics

In agent-based dynamics, each agent (i.e., node) interacts only with physically-closed nodes, called neighbors, rather than all other agents in replicator dynamics. In DTNs, nodes within the transmission range of a node can be regarded as neighbors of the node. Each node decides its behavior (a strategy) in the next round based on the information obtained in the preceding round. Agent-based dynamics reveals how the strategies, which are determined from local interactions, affect the performance of the whole system.

In every round, each node determines its strategy by comparing its own payoff with that of a randomly chosen neighboring node at the preceding round. Note that there is no assumption on the initial distribution of strategies. As we will see later, the initial strategy distribution almost does not have any influences on the system performance, except that it slightly affects the convergence time to the expected equilibrium of x_1^* discussed in Sect. 2.6.3. The strategy update of node u is conducted in the following probabilistic manner, called betters-possess-chance [19, 60]. At the beginning of each round, node u randomly chooses one of neighboring nodes, say, node v. If the average payoff Q_v of node v is greater than the average payoff Q_u of node u, node u then imitates the strategy of node v with probability $H(u, v)$.

$$H(u, v) = \frac{Q_v - Q_u}{T - P}, \tag{2.19}$$

where, $T - P$ $(= b)$ represents the maximum payoff difference. Otherwise, node u does not change its strategy. Thus, the more a strategy acquires the payoff, the more it spreads over the network through the imitation process in a hop-by-hop manner.

2.6.2 Simulation Model

Simulation experiments were conducted with NetLogo [4], a multi-agent programmable modeling simulator. For simplicity, we assume that the duration of a round is fixed and each node periodically generates a fixed number of bundles per round. Therefore c and s are constant and set accordingly as discussed in section 2.5. Mutation updating is introduced only in subsection 2.6.3.9. In the following figures,

the average of 100 independent simulation experiments is plotted.

2.6.3 Simulation Results

We first confirm the range of the number N of cluster members to which the prediction through replicator dynamics is applicable. After that, we discuss system characteristics in detail: The transient behavior, the role transitions of nodes, the effect of topological structures, the battery life of nodes, and successful bundle transfer characteristics.

2.6.3.1 System size valid for replicator dynamics

Figure 2.13 compares the analytical results of replicator dynamics on graphs with the simulation results of agent-based dynamics, where graphs are regular. We first conduct the experiments for no-retransmission case. Here b is set to be 1.67 and $c-s=1$. When both the number N of nodes and the degree k are large enough, agent-based dynamics attains the same equilibrium as predicted by replicator dynamics on graphs, because replicator dynamics on graphs assumes that $N=\infty$ and k is sufficiently large. When k is small, however, we observe a slight difference even for a large N. For example, when $N=100$, the equilibrium in the agent-based dynamics is at most 0.079 greater than that in the replicator dynamics. Contrarily, the results of agent-based dynamics for $N=10$ totally differ from those of replicator dynamics. Thus the number N of cluster members is essential in applying replicator dynamics to predicting the ratio of aggregators in equilibrium. In what follows, N is set to be 100.

2.6.3.2 Transient behavior

Figure 2.14 shows how the ratio x_1^* of aggregators converges to the equilibrium, where graphs are regular. We observe that x_1^* converges after 20 rounds for all cases. This quick convergence property is suitable for achieving a stable system. The resulting equilibrium is not greater than the predicted x_1^* in general and it coincides with x_1^* in the full mesh case, as shown in Figure 2.13.

Next, we investigate the influence of the initial strategy distribution on the convergence property. Recall that the predicted equilibrium x_1^* by the replicator dynamics is almost globally stable, i.e., if the initial value of x_1^* is in $(0,1)$, the replicator equation in (2.6) converges to the equilibrium x_1^* in Eq. (2.7). Therefore we expect that the agent-based dynamics inherits the stable convergence property. Although the

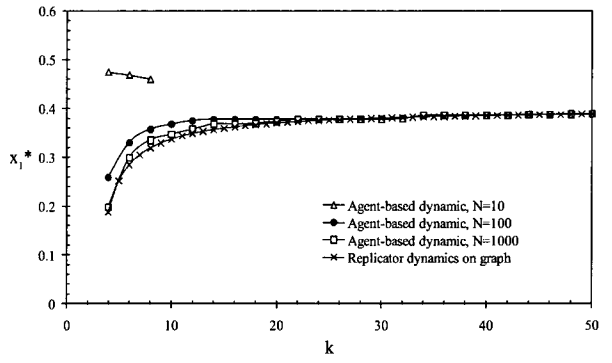


Figure 2.13: Equilibrium x_1^* in k-regular graphs (No-retransmission case, $b = 1.67$, $c - s = 1$).

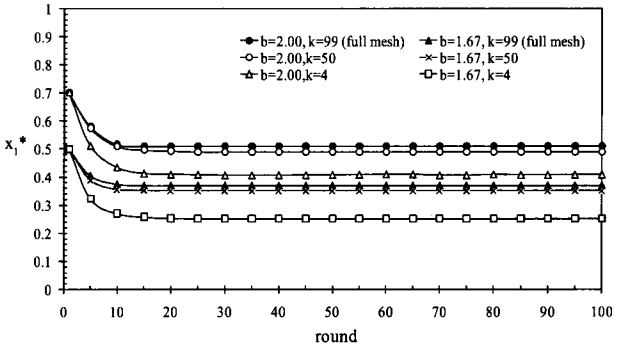


Figure 2.14: Transient behavior of ratio x_1^* in k-regular graphs (No-retransmission case, $N = 100$, $c - s = 1$).

convergence time depends on the initial value of x_1^*, we found that x_1^* converges to the same equilibrium after at most 20 rounds.

2.6.3.3 Role transition of nodes

We showed that the ratio of aggregators quickly converges to the equilibrium. The role of each node, however, is not fixed but it alternates dynamically over rounds, because each node selects its own strategy in a probabilistic manner. Figure 2.15 illustrates the probability of being an aggregator of node i, p_i ($i = 1, 2, \ldots, 100$), in 3,000 rounds, where nodes are sorted in ascending order of p_i. Note that the average $\overline{p}$ of p_i is equal to 0.2508 and the standard deviation of that is equal to 0.2000. The role transition contributes to load balancing and robustness against node failures.

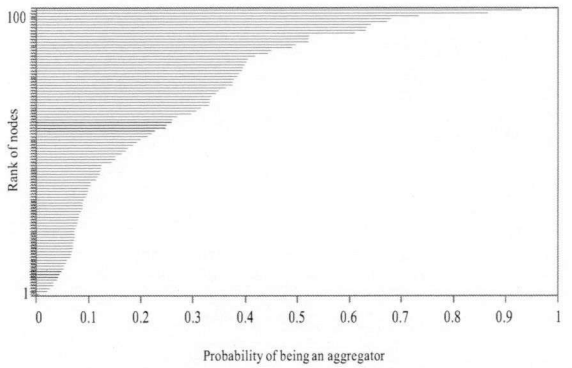


Figure 2.15: Probability of each node being an aggregator in a k-regular graph (No-retransmission case, $k = 4$, $b = 1.67$, $c - s = 1$).

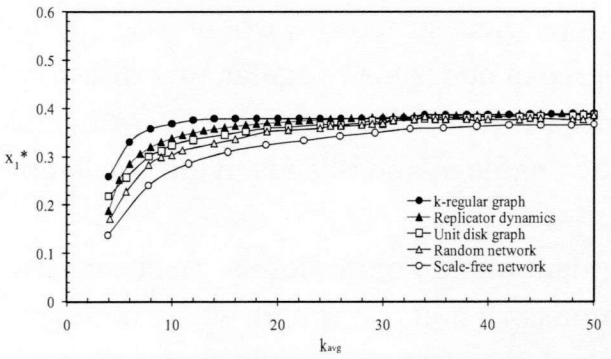


Figure 2.16: Influence of network topology on the equilibrium x_1^* (No-retransmission case, $b = 1.67$, $c - s = 1$).

2.6.3.4 Effect of topological structures

So far we have shown the simulation results with k-regular graphs. We now consider unit disk graphs which are more realistic networks and are suitable for abstracting wireless networks. The unit disk graphs are generated by randomly located nodes in 2-dimentional space where two nodes are adjacent if the transmission ranges of the nodes mutually cover each other. By setting the number N of nodes to be 100 we set the area size to be 1 x 1 [km^2], and the transmission range of each node is set to be 100 [m] in default. We additionally produce two famous network topologies: Scale-free networks and random networks with Barabasi-Albert (BA) model [7] and Erdos-Renyi (ER) model [14], respectively. Note that we can control the average degree k_{avg} by adjusting parameters in those models adequately. With those network models, we discuss the influence of topological structures on the system performance.

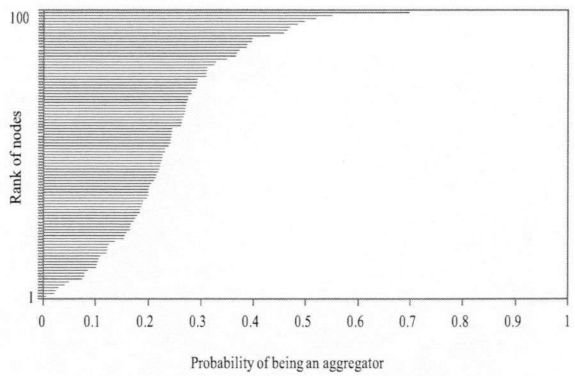


Figure 2.17: Probability of each node being an aggregator in a unit disk graph (No-retransmission case, $k_{\text{avg}} = 3.96$, $b = 1.67$, $c - s = 1$).

Figure 2.16 shows the equilibrium x_1^* as a function of average degree k_{avg} in networks with different topological structures, where $N = 100$, $b = 1.67$ and $c - s = 1$. The variance in the degree of nodes for k-regular, unit disk graph, random networks, and scale-free networks are 4.00, 4.73, 5.62 and 9.02, respectively. We observe that the large variance in the degree of nodes leads to the small ratio of aggregators x_1^* in equilibrium.

To investigate this phenomenon more closely, we observe two figures. Figure 2.17 shows p_i over 3,000 rounds in a unit disk graph, where nodes are sorted in ascending order of p_i. $\overline{p}$ is equal to 0.2112 and the standard deviation of p_i is equal to 0.1247. Compared with Figure 2.15 in a k-regular graph, we observe that $\overline{p}$ becomes small in the unit disk graph. Figure 2.18 is a scatter graph showing the degree d_i and p_i of node i ($i = 1, 2, \ldots, 100$) in a unit disk graph. We observe that the positive correlation between those two quantities; nodes with high degrees are likely to have large probabilities. In fact, the overall average probability $\overline{p}_W$ of being an aggregator weighted by node degree is equal to 0.2933, where

$$\overline{p}_W = \frac{\sum_{i=1}^{N} d_i p_i}{N k_{\text{avg}}},$$

which is greater than the un-weighted average probability $\overline{p}$. Thus we conclude that nodes with large degrees have a stronger impact in playing games than those with small degrees.

Next, we investigate the speed of the convergence to the equilibrium. Figure 2.19 shows the transient behavior of the ratio x_1^* of aggregators in k-regular graph, unit disk graph, random, and scale-free network with $N = 100$ and $b = 1.67$, where the average degree k_{avg} is set to be almost the same. We observe that it takes a longer time

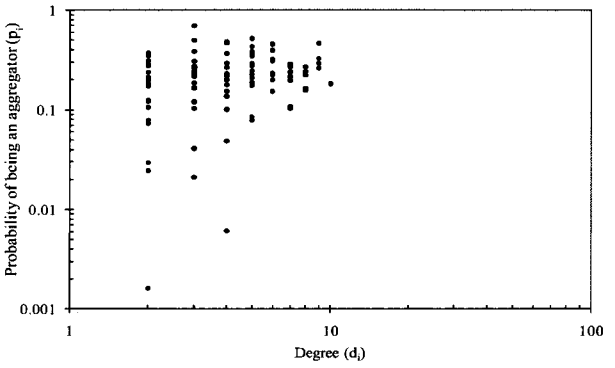


Figure 2.18: Degree vs. probability of each node being an aggregator in a unit disk graph (No-retransmission case, $k_{\text{avg}} = 3.96$, $b = 1.67$, $c - s = 1$) (log scale).

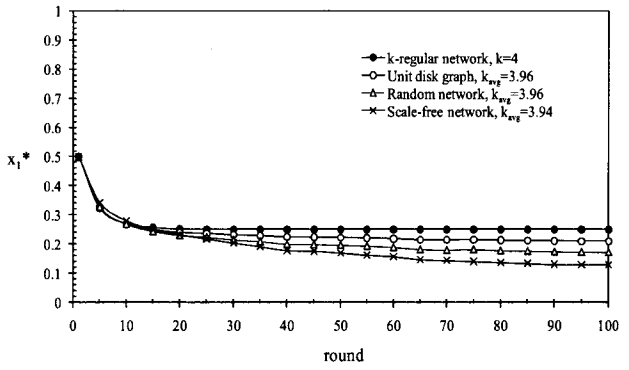
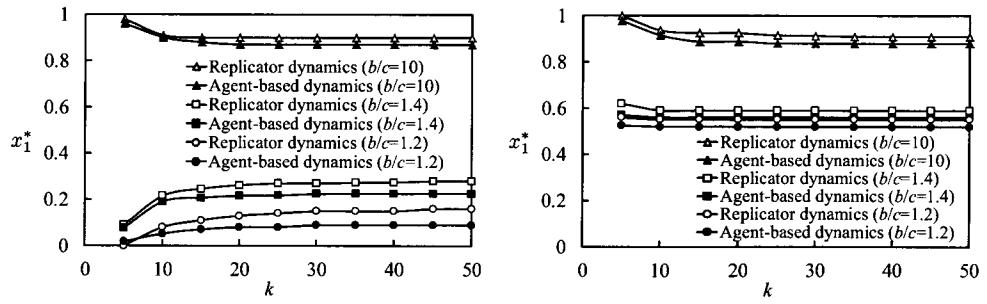


Figure 2.19: Transient behavior of ratio x_1^* in different topological structures (No-retransmission case, $b = 1.67$, $c - s = 1$).

for networks with high degree variation, compared with networks with low degree variation, yet 100 rounds is enough to converge to the equilibrium in all cases. In summary, the proposed scheme works well in those kinds of non-regular networks.

2.6.3.5 Consistency between replicator dynamics and agent-based dynamics

In what follows, we present the results of simulation experiments for both no-retransmission case and retransmission case. Figure 2.20 illustrates the equilibria x_1^* in replicator dynamics and agent-based dynamics, where the parameter sets are chosen in such a way that both the stability and running conditions are satisfied. We observe that x_1^* in the agent-based dynamics is slightly smaller than x_1^* in the replicator dynamics. These differences come from a relatively small system scale ($N = 100$) and a diversity of node degrees in the unit disk graphs. We conclude that the analytical result on the equiribria can grasp the general tendency of the system behavior.



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

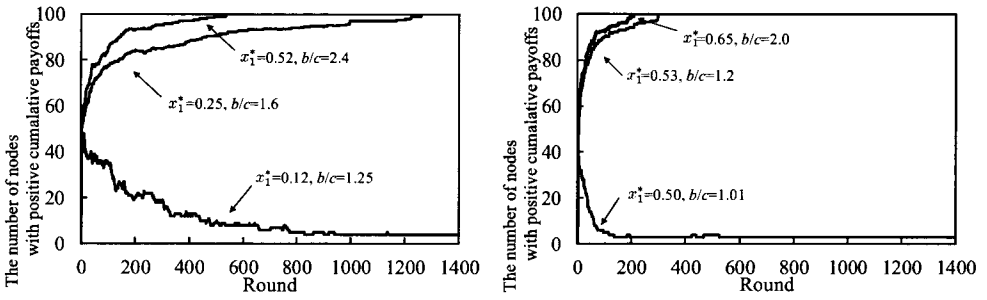
Figure 2.20: Validity of the theoretic analysis ($c = 100$, $s = 1$).

2.6.3.6 Battery life

Next, we examine the validity of the running condition given by Eq. (2.11). As discussed in Section 2.4.2, all nodes can survive forever under appropriate values of the parameters b/s and c/s. Figure 2.21 shows the transition of the number of nodes with positive cumulative payoffs for different x_1^*. Note that every node initially has no payoff. Each node obtains energy supply from the message ferry when it serves as an aggregator and has at least one neighboring node being a sender. Our aim is to achieve all nodes having positive cumulative payoffs, so that they can work permanently if they have a sufficient amount of initial battery. Given $c = 100$, $s = 1$, and $k = 5$, the infimum b^*/c of b/c satisfying Eq. (2.11) is given by 1.175 in the no-retransmission case and it is given by 1.009 in the retransmission case. As shown in Figure 2.21, we observe that all nodes have positive cumulative payoffs when b/c is enough large. On the other hand, when b/c is larger than yet close to b^*/c, many nodes have negative cumulative payoffs. This phenomenon can be explained as follows. Recall that the infimum of the equilibria x_1^* in the agent-based dynamics is smaller than the infimum of x_1^* in the replicator dynamics. As indicated in Eq. (2.12), the corresponding b^* in the agent-based dynamics would be greater than b^* in the replicator dynamics. As a result, if we set b in the agent-based dynamics to be close to b^* in the replicator dynamics, the agent-based system might fail to satisfy the running condition.

2.6.3.7 Successful bundle transfer

For senders (resp. aggregators), it is desirable that at least one aggregator (resp. sender) exists among neighboring nodes for successful bundle transfer. We define *sender* (resp. *aggregator*) *success probability* as the probability that senders (aggregators) have at least one neighboring node being an aggregator (resp. sender). These



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

Figure 2.21: Transition of the number of nodes with positive cumulative payoffs ($c = 100$, $s = 1$, k=5).

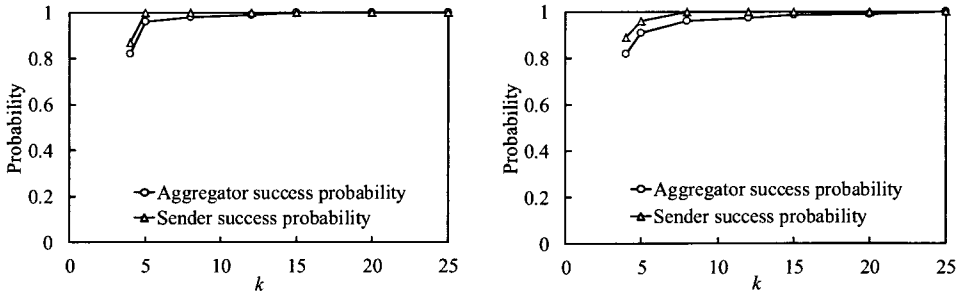
success probabilities are affected not only by b/s and c/s, but also by k. Figure 2.22 depicts these probabilities as functions of k. We observe that in any case, both sender and aggregator success probabilities go to one as $k \to \infty$. The reason is that each node has many neighbors on average when k is large.

Comparing Figs. 2.22 (b) and 2.22 (c), we also observe that when k is small, x_1^* has a positive correlation with those success probabilities. Recall that small x_1^* is preferable, but these results indicates that small x_1^* and k do not necessarily yield the success probabilities close to one. To clarify this, Figure 2.23 illustrates the minimum k that yields both probabilities over 0.9, as functions of x_1^*, where b is set adequately. We observe that the minimum k increases as x_1^* decreases but k is kept relatively low. This can be confirmed from the fact that k should be greater than $1/x_1^*$ for senders to have at least one aggregator in their neighbors.

2.6.3.8 Optimality of aggregator selection

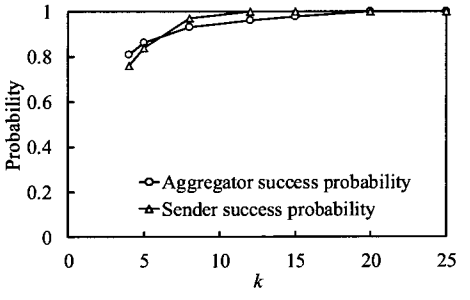
If a topological structure is given, there exists an aggregator placement with the minimum number of aggregators, which satisfies both aggregator and sender success probabilities become one. We regard such an aggregator placement as optimal. The optimal aggregator placement can be obtained as the solution of the following optimization problem P.

$$\begin{aligned} P: \quad & \text{minimize} \sum_{i \in \mathcal{N}} s_i, \\ & \text{subject to } s_i = \{0,1\}, \qquad \forall i \in \mathcal{N}, \\ & \qquad s_i + \sum_{j \in \mathcal{N}_i} s_j > 0, \quad \forall i \in \mathcal{N}, \end{aligned}$$



(a) Retransmission case (x_1^*=0.53 when k=5, b/c=1.2).

(b) No retransmission case (x_1^*=0.52 when k=5, b/c=2.1).



(c) No retransmission case (x_1^*=0.25 when k=5, b/c=1.3).

Figure 2.22: Sender/aggregator success probability satisfying stability and running conditions ($c = 100$, $s = 1$).

where $\mathcal{N}$ denote the set of nodes in the system and $\mathcal{N}_i$ denotes the set of neighboring nodes of node i. Furthermore, s_i ($i \in \mathcal{N}$) represents an indicator function of node i being an aggregator; it takes one if node i is an aggregator, and otherwise it takes zero.

We examine the optimality of aggregator selection given by the proposed scheme. Figure 2.24 illustrates the relationship between x_1^* and success probabilities for no-retransmission case when $c = 100$, $s = 1$, and $k = 10$. We also show the quasi-optimal result ($x_1^* = 0.14$), which is obtained by a greedy algorithm described in Algorithm 2.1. We observe that the aggregator success probability of the proposed scheme is almost equal to one regardless of x_1^*. On the other hand, the sender success probability of the proposed scheme is about 0.82 when $x_1^* = 0.14$ and it increases with x_1^*. From these results, we conclude that the proposed scheme cannot achieve the optimal aggregator placement but it can sufficiently distribute aggregators over the network.

2.6.3.9 Mutation updating

As mentioned in subsection 2.3.3, imitation updating has one drawback: When a node does not find any neighbors taking the different strategy, it cannot change the

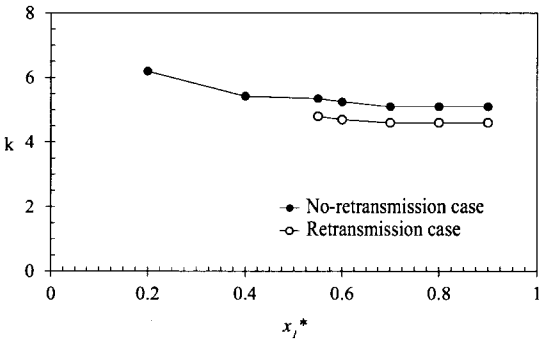


Figure 2.23: Minimum k that satisfies sender/aggregator success probability over 0.9, and stability and running conditions ($c = 100$, $s = 1$).

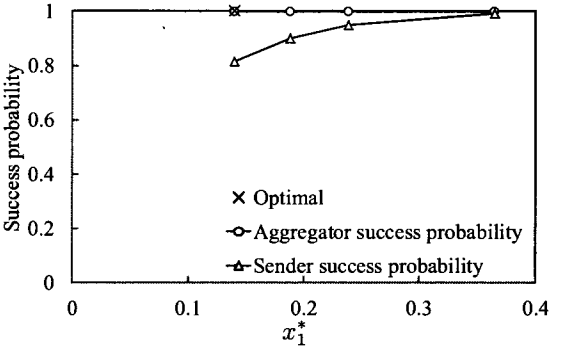


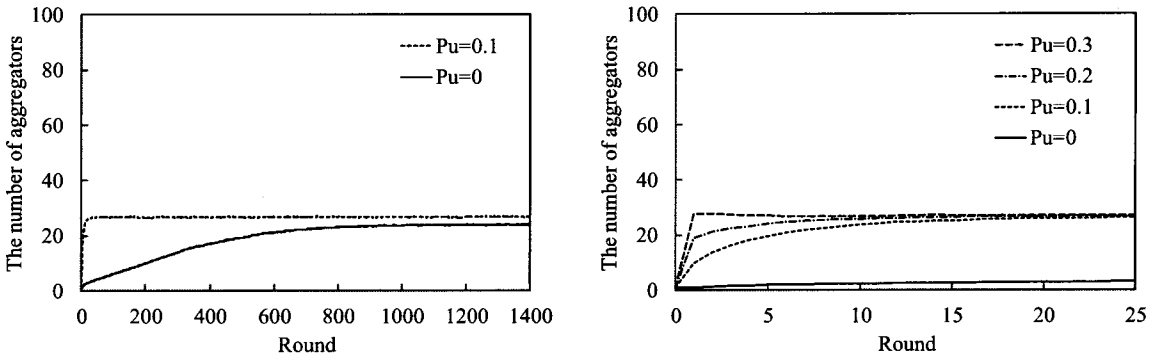
Figure 2.24: Optimality of aggregator selection (No-retransmission case, $c = 100$, $s = 1$, $k = 10$).

strategy. This feature will lead a slow convergence to the steady state. Moreover, this feature becomes problematic when there are few aggregators in the initial state. In such a situation, all nodes may choose to be senders in the next round and once this happens, the system will break down eventually. This can be avoided with mutation updating, where each node changes its strategy with probability P_u when all neighboring nodes take the same strategy as the node. Note that mutation updating will work when all adjacent aggregators disappear due to node failures, and therefore the resilience to node failures is expected to be improved with mutation updating.

In what follows, we examine extreme situations where only one aggregator exists in the initial state (i.e., round 0). Figure 2.25(a) depicts the change of the number of aggregators. We observe that the proposed scheme without mutation updating ($P_u = 0$) can reach the steady state in this example, even though the convergence is slow, compared with the system with $P_u = 0.1$. Introducing mutation updating ($P_u = 0.1$) dramatically improves the speed of convergence, while x_1^* increases slightly.

Algorithm 2.1: Heuristic algorithm for Problem P.

1: // *Initialization*
2: **for all** $i \in \mathcal{N}$ **do**
3: $s_i = 0$
4: **end for**
5: // *Aggregator placement*
6: **while** $\{i \mid s_i = 0, \sum_{j \in \mathcal{N}_i} s_j = 0\} \neq \phi$ **do**
7: $l = \arg\max_{j \in \{i \mid s_i = 0, \sum_{j \in \mathcal{N}_i} s_j = 0\}} |\mathcal{N}_j|$
8: $s_l = 1$
9: **end while**



(a) The number of aggregators.

(b) The number of aggregators in the first 25 rounds.

Figure 2.25: The transition of the number of aggregators (No-retransmission case, $b/c = 1.5$, $c = 100$, $s = 1$, $k = 10$, $x_1^* = 0.27$).

We further examine an appropriate setting of P_u. Figure 2.25(b) illustrates the change of the number of aggregators in the first 25 rounds. Since there is only one aggregator in round 0, most of the nodes adopt mutation updating with probability P_u. As a result, the number of aggregators steeply increases to about P_u in the first round. In the subsequent rounds, imitation updating dominates the system dynamics and the system reaches the steady state. This indicates that setting P_u to be x_1^* in Eq. (2.8) would be suitable for maximizing the convergence speed.

2.7 Conclusion

This chapter considered data aggregation in a cluster for ferry-assisted multi-cluster DTNs. We considered that nodes were inherently selfish and non-cooperative in nature. Applying evolutionary game theory, we proposed the self-organized data aggregation scheme in such an environment. In this scheme, the selection of aggrega-

tors is conducted through decentralized processes with the help of strategic decisions of evolutionary game theory. The proposed scheme was evaluated and we showed the excellent performance of the proposed scheme. In particular, we can control the numbers of the aggregators by setting parameters adequately.

Note that the controllable and stable equilibrium of the ratio of aggregators follows from the fact that a strategy different from the opponent yields a larger payoff. Therefore our proposed scheme also works well under such a situation that senders need to retransmit bundles when their transmissions fail and therefore they should awake until their successful transmissions. Taking account of this characteristics, we also examined the characteristics by taking account of bundle retransmission when a sender cannot find an aggregator as its neighbor. Through the analysis of replicator equation on graphs, we showed that the no-retransmission case is more suitable for achieving lower x_1^* with slightly higher b/c compared with the retransmission case.

In addition, we discussed running condition where all nodes can survive without battery outage which is important for permanently living systems. Moreover, we considered mutation updating and examined its impact.

To evaluate the validity of theoretic analysis of replicator dynamics and reveal feasible parameter settings achieving successful bundle transfer, we conducted simulation experiments using agent-based dynamics. Through simulation experiments, we obtained the following characteristics. (i) The analytical results grasp the general tendency of the agent-based dynamics in the unit disk graph, even though they assume a regular graph. (ii) Both sender and aggregator success probabilities can be over 0.9 even under relatively small k. (iii) The proposed scheme can achieve well-distributed placement of aggregators in a fully distributed manner. (iv) The mutation updating can improve the convergence speed. Both theoretic and simulation results presented appropriate parameter settings to achieve a system with desirable characteristics: Stability, survivability, and success probability in bundle transfer.

CHAPTER 3

Optimal Visiting Order of Isolated Clusters

THIS chapter focuses on one part of inter-cluster communication by studying the optimal visiting order of isolated clusters. Recall that in our proposed scenario, the fixed base station, i.e., sink node serves as a connector to the Internet or to other sink nodes. Where a message ferry helps the inter-cluster communication by acting as a mediator between each cluster and the outer world via the sink node, as shown in Figure 3.1. In such scenario, the problem is to find an efficient route along which the message ferry visits isolated clusters and the sink node.

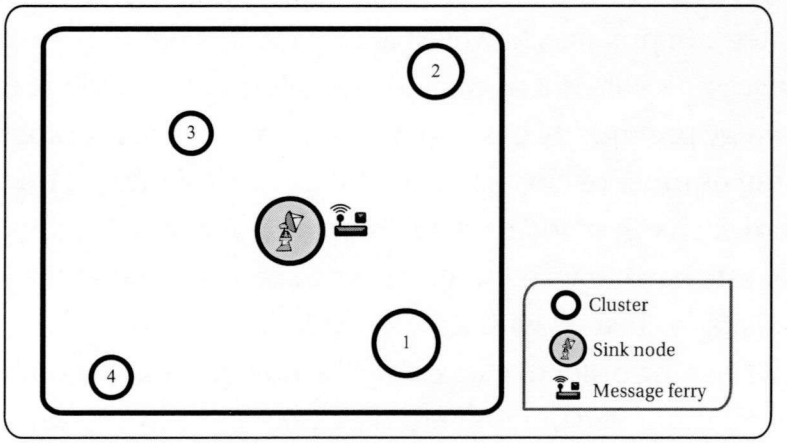


Figure 3.1: Proposed scenario.

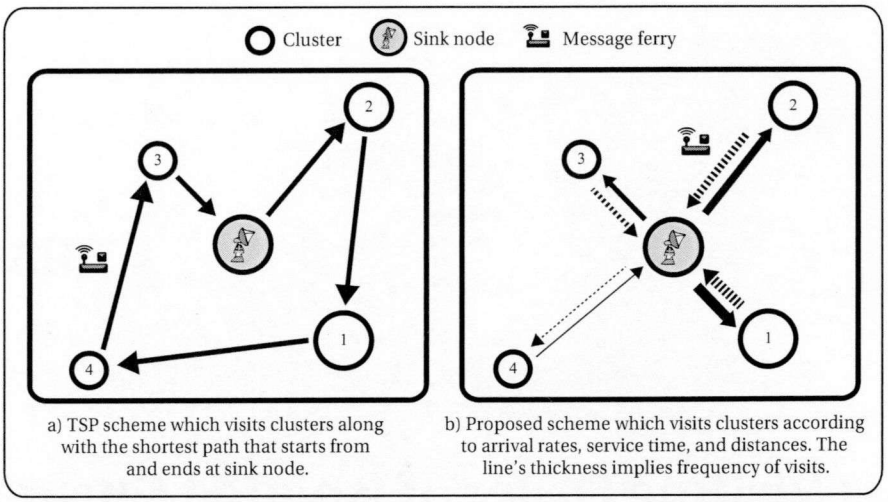


Figure 3.2: Example of message ferry's visiting sequence in TSP-based routing and the proposed scheme. Each arrow indicates the movement of message ferry and the size of each cluster is proportional to the arrival rate of bundles.

3.1 Optimal Visiting Order of Isolated Clusters to Minimize the Total Mean Delivery Delay of Bundles

Suppose service times (i.e., times needed for collecting bundles from clusters and unloading them to the sink node) are negligible. In such a case, the shortest cyclic route seems to be a natural solution, which can be obtained by solving the traveling salesman problem (TSP) [30]. The shortest cyclic route starts from the sink node, passes through each cluster at once, and finally returns the sink node as shown in Figure 3.2(a). Therefore in terms of the mean waiting time, all clusters are treated fairly in this strategy. In practice, however, arrival rates of bundles are different among clusters and service times are not negligible. In such situations, the TSP-based shortest cyclic route strategy potentially has two drawbacks: 1) The time spent for one cycle increases with the number of clusters, and 2) if the arrival rates of bundles at clusters are different from each other and service times are not negligible, bundles in clusters with high arrival rate have to wait for long time to be delivered to the sink node while less important visits to clusters with a few bundles also take place.

In general, the visiting order of clusters by the message ferry should be determined based on arrival rates, service times of bundles, and one-way traveling times between clusters and the sink node. We assume that all the isolated clusters are significantly apart from each other. As a result, the *inter-visit time* of a cluster (i.e., the interval time between a departure of the message ferry from the cluster and the next return of the message ferry to that cluster) naturally becomes long. In such a situation, when

the message ferry visits each cluster, it would find bundles that wait for a long time with high probability. Therefore, in order to reduce the delivery delay, it might be reasonable to deliver those bundles to the sink node directly, as shown in Figure 3.2(b), rather than to visit other clusters while carrying them.

This inter-cluster communication of the message ferry can be best studied using a polling model [52, 53],

Proposed Method: ***Polling Model:*** In the polling system, the message ferry, clusters, and bundles are regarded as the server, stations, and customers, respectively, and "service" means that the message ferry collects (unloads) bundles from (to) the cluster (sink node). The optimization of the polling order are studied in [8, 9, 31], which is equivalent to find an optimal visiting order of stations, which minimizes the expected waiting time of all customers.

In the study of polling models, the waiting time (i.e., the length of an interval from the generation of a bundle to the instant at which its service starts) is a primary performance measure of interest. On the other hand, in our system, we are interested in the *delivery delay*, which is defined as the time interval from the generation of a bundle to the completion of its delivery to the sink node. In this chapter, we show that the mean delivery delay of bundles is given in terms of the weighted sum of the mean waiting times of bundles at respective clusters. We then apply the optimization technique in [8, 10, 43] to our system and obtain a quasi-optimal visiting order that minimizes the total mean delivery delay of the system. Roughly speaking, clusters with high arrival rate and/or close to the sink node are visited more frequently than others in the optimal visiting order.

The rest of this chapter is organized as follows. In Section 3.2. we review the related work. We describe the mathematical model in Section 3.3. Sections 3.4 provides the optimization problem formulation and its solution method. Section 3.5 shows the result of simulation experiments and demonstrates the effectiveness of our scheme. Finally we conclude the chapter in Section 3.6.

3.2 Related Work

Zhao et al. first applied the TSP-based routing to highly-partitioned ad hoc wireless networks [63, 64] by introducing a message ferry as the traveling salesman. A single ferry is used for communications among fixed nodes in partitioned networks [63, 64] and a heuristic method for finding the visiting order is shown. In [63], they also extended their message ferry scheme to that for systems with mobile nodes.

In [6], Ammar et al. focused on the buffer size required for each node when the message ferry travels along the shortest cyclic path. They presented an algorithm for finding the visiting order that minimizes the maximum required buffer size among nodes. This problem can be regarded as a variant of the TSP problem under the assumptions of identical arrival rate and negligible service time, and minimizing the buffer size is equivalent to minimizing the mean waiting time for the ferry visiting. The objective is similar to ours but this approach is not suitable for scenarios with heterogeneous arrival rate and non-negligible service time.

Some works tried to improve the scalability and robustness of the system with the help of multiple message ferries, e.g., multiple ferries for a single route [6] and multiple ferries for multiple routes [65]. They considered the message ferry assignment to nodes and route making in such a way that the number of message ferries is minimized when the number of nodes and the upper bound of the waiting time are given. Miura et al. considered clustering of highly-partitioned wireless networks [34]. They assume that there are several partitioned clusters in which physically-close nodes exist; which is similar to our scenario in Figure 4.1. They applied the TSP-based routing by setting the visiting point of the message ferry to the center of each cluster.

All of the above mentioned studies assume that arrival rates are identical among nodes and service times are negligible. In practical situations, however, these assumptions do not necessarily hold. In such situations, finding the shortest cyclic path is insufficient to achieve minimizing the overall mean delivery delay of bundles. Kavitha et al. first tackled this problem by applying the polling model. In [25–27], they assumed message ferry-based wireless LANs, where nodes are well scattered over the area and designed an optimal route (among some given class of trajectories, e.g., circle and line) that minimizes the overall expected waiting times. The message ferry can serve nodes within its transmission range at any point on the path. Their approach can also support both uplink and downlink services.

Although our objective is similar to [25–27], the target scenario is totally different. It is assumed in [25–27] that nodes can exist at any point in an area according to a known probability distribution, while we assume that there are partitioned clusters, each of which consists of physically-close nodes. If the approach in [25–27] is applied to our scenario, it requires many paths to cover the whole area, each of which is a circle/line trajectory supported by a single message ferry. In addition, in [25–27] a cyclic policy is used: The server visits the stations in a predetermined cyclic order. Hence, if clusters with high arrival rates and those with low arrival rates coexist in the area, it will not be effective. On the other hand, our proposed scheme applies a non-cyclic policy, taking account of the arrival rate and location of each cluster.

3.3 Model

Suppose the system consists of N clusters labeled 1 to N, the sink node, and a message ferry, all of which have buffers of infinite capacity. The message ferry periodically visits clusters according to a predefined visiting order (i.e., a polling table). When the message ferry arrives at a cluster, it serves bundles under the exhaustive service discipline, i.e., bundles are transmitted successively to the message ferry, and when there are no waiting bundles, the message ferry leaves the cluster. It is known that the exhaustive service discipline has the best performance in terms of the overall mean waiting time [52]. After collecting all bundles at the cluster, the message ferry immediately returns to the sink node, unloads all bundles it carries to the sink node, and goes to the next cluster.

We define S_i ($i \in \mathcal{N}$) as the one-way traveling time between cluster i and the sink node, where $\mathcal{N} = \{1, 2, \ldots, N\}$. We assume that S_i ($i \in \mathcal{N}$) is constant because of the fixed physical route and the constant speed of the message ferry. Bundles at cluster i ($i \in \mathcal{N}$) are generated according to a Poisson process with rate λ_i and all of them are stored at cluster i. Service times X_i ($i \in \mathcal{N}$) of bundles at cluster i follow a general distribution with finite mean x_i and second moment $x_i^{(2)}$. Note that X_i corresponds to the transmission time of a randomly chosen bundle at cluster i. We assume that high speed channels are available at the sink node, and therefore the unloading time of bundles at the sink node is assumed to be negligible.

Let $\rho_i = \lambda_i x_i$ ($i \in \mathcal{N}$) denote the traffic intensity at cluster i. The overall generation rate of bundles and the overall traffic intensity are denoted by $\lambda = \sum_{i \in \mathcal{N}} \lambda_i$ and $\rho = \sum_{i \in \mathcal{N}} \rho_i$, respectively. We assume that $\rho < 1$, which ensures the stability of the system [52]. In what follows, the system is assumed to be in steady state.

3.4 Optimization problem formulation and its solution method

We define the delivery delay of bundles as the time interval from the generation of the bundle to the instant at which it is delivered to the sink node. Let $W_{\text{deliver},i}$ ($i \in \mathcal{N}$) denote the delivery time of a randomly chosen bundle generated at cluster i. The goal of this section is to formulate and solve a mathematical program to find the optimal visiting order of clusters, which minimizes the overall mean delivery delay $\mathrm{E}[W_{\text{total}}]$:

$$\mathrm{E}[W_{\text{total}}] = \sum_{i \in \mathcal{N}} \frac{\lambda_i}{\lambda} \mathrm{E}[W_{\text{deliver},i}]. \tag{3.1}$$

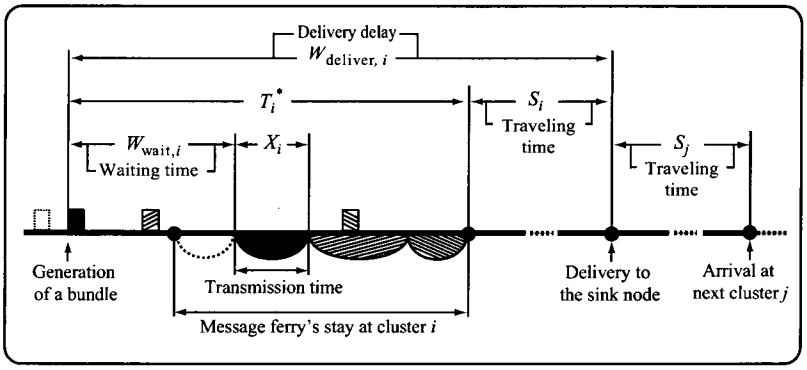


Figure 3.3: Timing chart (exhaustive service policy). When the message ferry arrives at cluster i, there are already three bundles waiting for the service. During the service for them, one bundle is further generated. When there is no bundle to be served, the message ferry leaves cluster i and visits the next cluster j via the sink node.

As we will see, our problem is reduced to the minimization problem of a weighted sum of mean waiting times of a polling model. Without loss of generality, we assume that bundles at each cluster are served on an first-come, first-served (FCFS) basis, because $\mathrm{E}[W_{\text{deliver},i}]$ $(i \in \mathcal{N})$ is irrelevant to the service order of waiting bundles at cluster i in the exhaustive service discipline.

We first divide $W_{\text{deliver},i}$ $(i \in \mathcal{N})$ into two disjoint parts T_i^* and S_i, where T_i^* denotes the sojourn time of a randomly chosen bundle at cluster i. See Figure 3.3. It then follows that

$$\mathrm{E}[W_{\text{deliver},i}] = \mathrm{E}[T_i^*] + S_i, \quad i \in \mathcal{N}. \tag{3.2}$$

In the exhaustive service policy, the message ferry has to stay at each cluster until it finishes collecting all bundles. Therefore $\mathrm{E}[T_i^*]$ is considered as the mean delay cycle with an initial delay $W_{\text{wait},i} + X_i$, where $W_{\text{wait},i}$ denotes the waiting time of a randomly chosen bundle at cluster i (see Figure 3.3). We then have [13]

$$\mathrm{E}[T_i^*] = \frac{\mathrm{E}[W_{\text{wait},i}] + x_i}{1 - \rho_i}. \tag{3.3}$$

Note that $W_{\text{wait},i}$ is identical to the waiting time in the ordinary polling model.

It then follows from Eqs. (3.1), (3.2), and (3.3) that

$$\begin{aligned} \mathrm{E}[W_{\text{total}}] &= \frac{1}{\lambda} \sum_{i \in \mathcal{N}} \lambda_i \left(\frac{\mathrm{E}[W_{\text{wait},i}]}{1 - \rho_i} + \frac{x_i}{1 - \rho_i} + S_i \right) \\ &= \sum_{i \in \mathcal{N}} c_i \mathrm{E}[W_{\text{wait},i}] + \alpha, \end{aligned} \tag{3.4}$$

where

$$c_i = \frac{\lambda_i}{(1-\rho_i)\lambda}, \quad i \in \mathcal{N}, \tag{3.5}$$
$$\alpha = \frac{1}{\lambda}\sum_{i\in\mathcal{N}}\left(\frac{\rho_i}{1-\rho_i} + \lambda_i S_i\right).$$

Because α is constant regardless of the visiting order of clusters, the minimization of $\mathrm{E}[W_{\mathrm{total}}]$ is equivalent to that of the weighted sum of the mean waiting times $\mathrm{E}[W_{\mathrm{wait},i}]$ in the exhaustive-service polling model:

$$\text{minimize} \quad \sum_{i\in\mathcal{N}} c_i \mathrm{E}[W_{\mathrm{wait},i}]. \tag{3.6}$$

In the rest of this section, we follow the lower bound approach in [8, 10], and obtain an approximate solution of Eq. (3.6).

Under the exhaustive service discipline, the mean waiting time $\mathrm{E}[W_{\mathrm{wait},i}]$ ($i \in \mathcal{N}$) at cluster i takes a form: [8]

$$\mathrm{E}[W_{\mathrm{wait},i}] = \frac{\lambda_i x_i^{(2)}}{2(1-\rho_i)} + \frac{v_i^{(2)}}{2v_i}, \quad i \in \mathcal{N}, \tag{3.7}$$

where v_i and $v_i^{(2)}$ ($i \in \mathcal{N}$) denote the first and second moments of interval lengths from departures of the message ferry from cluster i to the next arrival instants. Because $v_i^{(2)} \geq v_i^2$, the weighted sum of $\mathrm{E}[W_{\mathrm{wait},i}]$ is bounded from below:

$$\sum_{i\in\mathcal{N}} c_i \mathrm{E}[W_{\mathrm{wait},i}] \geq \frac{1}{2}\sum_{i\in\mathcal{N}} c_i \left(\frac{\lambda_i x_i^{(2)}}{1-\rho_i} + v_i\right). \tag{3.8}$$

We adopt the approach of [8] minimizing the lower bound given by the right hand side of Eq. (3.8), instead of $\sum_{i\in\mathcal{N}} c_i \mathrm{E}[W_{\mathrm{wait},i}]$.

Let q_i ($i \in \mathcal{N}$) denote the mean number of visits at cluster i per unit time. Because q_i^{-1} ($i \in \mathcal{N}$) is equal to the mean cycle time $\mathrm{E}[C_i] = v_i/(1-\rho_i)$ [52], we have

$$v_i = \frac{1-\rho_i}{q_i}. \quad i \in \mathcal{N}. \tag{3.9}$$

Substituting Eq. (3.9) into the right hand side of Eq. (3.8), rearranging terms with Eq. (3.5), and ignoring constant factors and terms, we obtain the objective function $f(\boldsymbol{q})$ of the minimization problem.

$$f(\boldsymbol{q}) = \sum_{i\in\mathcal{N}} \frac{\lambda_i}{q_i}, \tag{3.10}$$

where $\boldsymbol{q} = (q_1, q_2, \ldots, q_N)$.

The constraints on $\boldsymbol{q}$ are obtained as follows. First of all, $q_i > 0$ for all i ($i \in \mathcal{N}$). Furthermore

$$\rho + 2\sum_{j\in\mathcal{N}} S_j q_j = 1,$$

should hold. Note that $2S_j q_j$ ($j \in \mathcal{N}$) denote the time-average probability that the message ferry is traveling between the sink node and cluster i. Because ρ represents the probability of one of the clusters being served. Therefore the sum of them should be equal to one. In summary, we have the following Problem P.

$$\begin{aligned} P: \ & \text{minimize } f(\boldsymbol{q}), \\ & \text{subject to } \rho + 2\sum_{j\in\mathcal{N}} S_j q_j = 1, \\ & \qquad q_i > 0, \quad i \in \mathcal{N}. \end{aligned} \tag{3.11}$$

Problem P is easy to solve with the Lagrange multipliers method. We define $L(\boldsymbol{q}, \theta)$ as

$$L(\boldsymbol{q}, \theta) = f(\boldsymbol{q}) + \theta(\rho + 2\sum_{j\in\mathcal{N}} S_j q_j - 1),$$

where $\theta > 0$ denotes the Lagrange multiplier. We then have

$$\frac{\partial L}{\partial q_i} = -\frac{\lambda_i}{q_i^2} + 2\theta S_i = 0, \quad i \in \mathcal{N},$$

from which, it follows that

$$q_i = \sqrt{\frac{\lambda_i}{2\theta S_i}} > 0, \quad i \in \mathcal{N}. \tag{3.12}$$

q_i in Eq. (3.12) should satisfy Eq. (3.11), so that

$$\rho + \sqrt{\frac{1}{\theta}} \cdot \sum_{j\in\mathcal{N}} \sqrt{2\lambda_j S_j} = 1,$$

from which, it follows that

$$\sqrt{\frac{1}{\theta}} = \frac{1-\rho}{\sum_{j\in\mathcal{N}} \sqrt{2\lambda_j S_j}},$$

and therefore we obtain from Eq. (3.12)

$$q_i = \frac{1-\rho}{\sum_{j\in\mathcal{N}} \sqrt{2\lambda_j S_j}} \cdot \sqrt{\frac{\lambda_i}{2S_i}}, \quad i \in \mathcal{N}. \tag{3.13}$$

Let p_i ($i \in \mathcal{N}$) denote the ratio of the message ferry's visit to cluster i. It then follows from Eq. (4.2) that

$$p_i = \frac{q_i}{\sum_{j\in\mathcal{N}} q_j} = \frac{\sqrt{\lambda_i/S_i}}{\sum_{j\in\mathcal{N}} \sqrt{\lambda_j/S_j}}. \tag{3.14}$$

Eq. (3.14) indicates that the optimal frequency of visits to clusters is determined only by the ratio of the arrival rate λ_i ($i \in \mathcal{N}$) to the one-way travel times S_i ($i \in \mathcal{N}$), and it is independent of service times x_i ($i \in \mathcal{N}$). Thus, in our proposed scheme, the message ferry frequently visits clusters with high arrival rates and/or small distances to the sink node.

The next is to find the visiting order of clusters, whose frequency is given by Eq. (3.14). When non-periodic orders are allowed, this problem is called balanced sequence/words and examined in [5, 45], where each cluster is spaced as evenly as possible in the sequence. In our system, however, the target frequency p_i is an approximate one and the frequency of visits to each cluster is not exactly identical to the target frequency. Taking account of it, we use the following procedure for determining the visiting order of clusters, which is a combination of proposals in [10, 43].

Step 1: *Determination of the cycle length and the frequency of visits.* We borrow an idea in [10]. Let M denote an positive integer representing the cycle length in terms of the number of visited clusters. Also, let m_i ($i \in \mathcal{N}$) denote the number of visits to cluster i in a cycle. We define $\mathrm{int}(x)$ ($x > 0$) as

$$\mathrm{int}(x) = \begin{cases} \lfloor x \rfloor, & x - \lfloor x \rfloor < 0.5, \\ \lceil x \rceil, & \text{otherwise.} \end{cases}$$

For $m = N, N+1, \ldots$, we seek minimum $m = m^*$ such that

$$\mathrm{int}(m^* p_i) \geq 1, \quad i \in \mathcal{N},$$

$$|m^* p_i - \mathrm{int}(m^* p_i)| \leq \epsilon, \quad i \in \mathcal{N},$$

and

$$\sum_{i \in \mathcal{N}} \mathrm{int}(m^* p_i) = m^*,$$

where ϵ is a predetermined parameter. We then set

$$M = m^*, \qquad m_i = \mathrm{int}(m^* p_i) \quad (i \in \mathcal{N}).$$

Step 2: *Determination of the visiting order.* We use the procedure given in Appendix C of [43], which is summarized as follows. Let $\mathcal{M} = \{m_i;\ i \in \mathcal{N}\}$ denote the set of the numbers of visits to respective nodes in a cycle. For $r \in \mathcal{M}$, let $\mathcal{I}^{(r)} = \{i \in \mathcal{N};\ m_i = r\}$ denote the set of indices of clusters visited r times in a cycle. Furthermore, let $Q^{(r)}$ ($r \in \mathcal{M}$) denote a repeated string of symbols in $\mathcal{I}^{(r)}$, where each symbol appears r times with equal distance. For example, if $\mathcal{I}^{(3)} = \{2, 4\}$, $Q^{(3)}$ is given by 242424. For any string A, let $|A|$ denote the length of string A.

1. Prepare $Q^{(r)}$ for all $r \in \mathcal{M}$.

2. Choose $r \in \mathcal{M}$ and $\mathcal{D} = \{r\}$. Let $P = Q^{(r)}$.

3. If $\mathcal{M} \setminus \mathcal{D} = \emptyset$, stop the procedure, where P gives the visiting order of clusters.

4. Choose $r \in \mathcal{M} \setminus \mathcal{D}$ and $\mathcal{D} := \mathcal{D} \cup \{r\}$. We then merge $Q^{(r)}$ into P, and the resulting string is denoted by $P^{(r)}$, where $|P^{(r)}| = |P| + |Q^{(r)}|$. The rule of this merging operation is as follows. The kth symbol in $Q^{(r)}$ is identical to the $(k + d(k))$th symbol in $P^{(r)}$, where

$$d(k) = \mathrm{int}((k-1)|P|/|Q^{(r)}|), \quad k = 1, 2, \ldots, |Q^{(r)}|.$$

   The rest of symbols in $P^{(r)}$ is identical to those in P, and the order of those symbols are identical in P and $P^{(r)}$.

5. Let $P = P^{(r)}$ and go to step 3.

The flowchart of the overall procedures to obtain optimal visiting order is presented in Figure 3.4.

3.5 Simulation results

In this section, we evaluate the performance of our proposed scheme through simulation experiments.

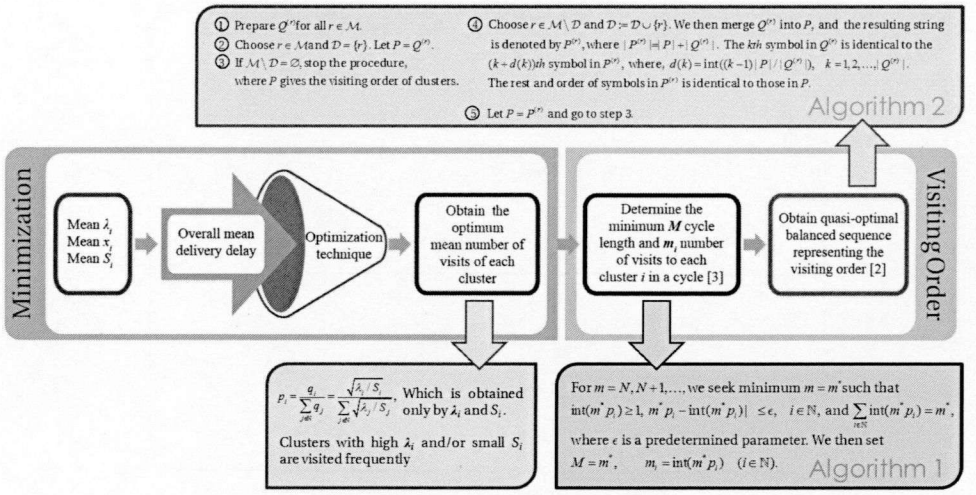


Figure 3.4: Flowchart of obtaining optimal vising order.

Table 3.1: Scenarios of λ_i (N=10, λ=0.76 [1/s]).

Case		λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_9	λ_{10}
Hetero.	Descend.	.30	.10	.08	.07	.06	.05	.04	.03	.02	.01
	Random	.02	.05	.30	.10	.08	.07	.04	.01	.03	.06
	Ascend.	.01	.02	.03	.04	.05	.06	.07	.08	.10	.30
Homogeneous		.076									

3.5.1 Simulation setting

We consider a system composed of a sink node and ten isolated clusters ($N = 10$). We use two kinds of cluster layouts: Circle-based layout and random layout models. In the circle-based layout model, ten clusters are placed equally dividing a circle with a radius of 13km, and the sink node is located at the center of the circle. On the other hand, the random layout model is illustrated in Figure 3.5. The circle-based layout and random layout models correspond to the cases of identical and different one-way traveling times S_i ($i = 1, 2, \ldots, 10$), respectively. We assume that the message ferry travels at a fixed speed of 10m/s (i.e., 36km/h). We denote the mean one-way traveling time by $\overline{S} = N^{-1} \sum_{j \in \mathcal{N}} S_j$, which is fixed to $1,300$ [s] in any case. Transmission times of bundles at all clusters are independent and identically distributed according to an exponential distribution with mean $x_i = 1$ [s]. For the settings of λ_i ($i = 1, 2, \ldots, 10$), we consider four cases, one is the homogeneous case and other three cases are heterogeneous, as shown in Table 3.1. In the following results, we mainly examine how λ_i and S_i affect the mean delivery delay $\mathrm{E}[W_{\text{total}}]$ (sec).

We compute the visiting order of clusters according to the procedure in Section 3.4, where ϵ is set to be 0.4. Recall that in the proposed visiting order, the message ferry

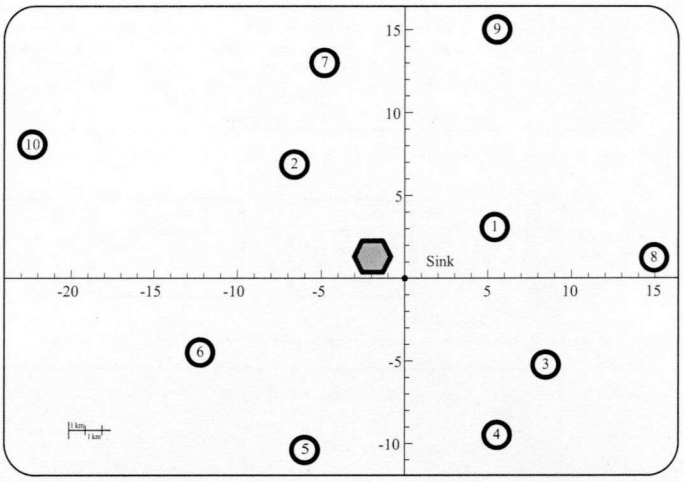


Figure 3.5: Random layout model ($N = 10$, S_1=600, S_2=900, S_3=1,000, S_4=1,100, S_5=1,200, S_6=1,300, S_7=1,400, S_8=1,500, S_9=1,600, S_{10}=2,400, $\overline{S} = 1,300$, $C = 13,304.94$). The cluster IDs are assigned in an ascending order of the distance from the sink node.

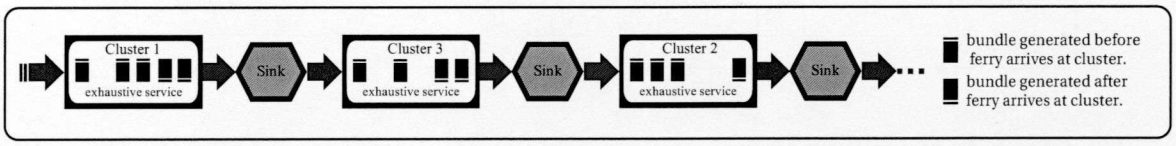


Figure 3.6: Message ferry's traveling sequence for the proposed optimal visiting order (for example the visiting order becomes 1–3–2).

returns to the sink node before visiting the next cluster, as shown in Figures 3.2 (b) and Figures 3.6 . For the sake of comparison, we also consider a cyclic visiting order and a TSP-based routing (cf. Figure 3.2 (a)). In the cyclic visiting order, the message ferry visits clusters one by one via the sink node, i.e., 1–sink–2–sink–$\cdots$. On the other hand, in the TSP-based routing, the message ferry visits clusters according to the shortest cyclic path that starts from and ends with the sink node. Let C denote the traveling time of one cycle in the TSP-based routing. We then have $C = 9,830.92$ and $C = 13,304.94$ in the circle-based and random layout models, respectively. For each simulation experiment, we discard the initial interval of 50,000 seconds as transient period and collect data in the subsequent interval from 50,000 to 6,000,000 (sec). All simulation results are presented with 95% confidence intervals, based on ten independent simulation runs.

Table 3.2: Mean delivery delay $\mathrm{E}[W_{\mathrm{total}}]$ in the circle-based layout model with homogeneous arrival rates (N=10).

	Visiting order	$\mathrm{E}[W_{\mathrm{total}}]$
Proposal	1-2-3-4-5-6-7-8-9-10-	44,679.25±87.26
Cyclic	1-2-3-4-5-6-7-8-9-10-	44,679.25±87.26
TSP	sink-1-2-3-4-5-6-7-8-9-10-sink-	38,290.08±49.84

3.5.2 Performance evaluation

We first evaluate the performance of three schemes in the circle-based layout, where S_i are homogeneous. Table 3.2 shows the visiting order and the mean delivery delay $\mathrm{E}[W_{\mathrm{total}}]$ when λ_i's are homogeneous. Note that our proposed scheme is identical with the cyclic scheme in this case because λ_i/S_i's ($i \in \mathcal{N}$) are identical. We observe that the TSP-based routing has the smallest $\mathrm{E}[W_{\mathrm{total}}]$ in this scenario.

Next, we examine the influence of the heterogeneity of λ_i. Table 3.3 shows the result in the circle-based layout, where λ_i's are set according to the descending/random/ascending arrival rate scenarios in Table 3.1. Note that the descending and ascending scenarios are both extremes and therefore in each scheme, the random arrival rate scenario yields the second best mean delivery delay. Even though the TSP-based routing has the smallest delivery delay, the difference between our proposed scheme and TSP-based routing becomes small, compared with the homogeneous case in Table 3.2.

The small difference of the results within our proposed scheme comes from a specific implementation of the procedure for generating the visiting order, where clusters are always arranged in an ascending order of their indices. If we arranged clusters in the descending order of their indices in the case of the ascending arrival rate scenario, we would have the visiting order of 10-9-8-7-10-6-5-9-4-10-3-2-1- and the result would be identical to that in the descending arrival rate scenario. The cyclic visiting order is essentially identical to the ordinary polling model, and the mean waiting time in asymmetric polling models is known to depend on the visiting order [18].

Recall that neither the TSP-based routing nor the cyclic visiting order take account of arrival rates at clusters. Compared with the cyclic visiting order, the difference between the mean delivery delay of the descending and ascending arrival rate scenarios in TSP-based routing is significantly large by the following reason. In the TSP-based routing, the message ferry visits clusters successively while carrying collected bundles with it, before returning to the sink node. In the descending arrival rate scenario, the message ferry tends to collect many bundles at clusters with small indices (i.e., in

Table 3.3: Mean delivery delay $E[W_{total}]$ in the circle-based layout model with heterogeneous arrival rates (N=10).

	λ_i	Visiting order	$E[W_{total}]$
Proposal	Descend.	1-2-3-4-1-5-6-2-7-1-8-9-10-	38,004.41±63.15
	Random	3-4-1-2-3-5-6-4-7-3-8-9-10-	38,142.16±38.95
	Ascend.	10-9-1-2-10-3-4-9-5-10-6-7-8-	38,246.43±55.56
Cyclic	Descend.	1-2-3-4-5-6-7-8-9-10-	45,353.64±61.46
	Random	1-2-3-4-5-6-7-8-9-10-	45,434.70±63.28
	Ascend.	1-2-3-4-5-6-7-8-9-10-	45,654.29±53.12
TSP	Descend.	sink-1-2-3-4-5-6-7-8-9-10-sink-	36,983.58±82.62
	Random	sink-1-2-3-4-5-6-7-8-9-10-sink-	36,384.11±33.85
	Ascend.	sink-1-2-3-4-5-6-7-8-9-10-sink-	33,744.17±68.49

Table 3.4: Mean delivery delay $E[W_{total}]$ in the circle-based layout model with one heavily loaded cluster ($N = 10$, $\lambda = 0.76$, $\lambda_1 = 0.9\lambda$, $\lambda_i = 0.1\lambda/9$ ($i = 2, 3, \ldots, 10$)).

	Visiting order	$E[W_{total}]$
Proposal	1-2-1-3-1-4-$\cdots$-1-9-1-10-	7072.23±39.45
TSP	sink-1-2-3-4-5-6-7-8-9-10-sink-	14901.43±67.89
	sink-10-9-8-7-6-5-4-3-2-1-sink-	8285.17±56.17

the former part of the cycle), and it carries them while visiting other lightly-loaded clusters with large indices. In this way, many bundles suffer from long delay, which leads to a significant increase of the mean delivery delay in the descending arrival rate scenario.

Note here that the TSP-based routing is not always superior to our proposed scheme. For example, suppose 90% of traffic is generated at cluster 1 and the rest is divided evenly among nine other clusters, while keeping the total traffic intensity fixed to λ=0.76. Table 3.4 shows the result. The mean delivery delay in the TSP-based routing is greater than that in our proposed scheme and in the TSP-based routing, the direction at which the message ferry moves affects the performance significantly.

We now turn our attention to the random layout model in Figure 3.5, where distances between clusters and between the sink node and respective clusters are not

Table 3.5: Mean delivery delay $E[W_{total}]$ in the random layout model with homogeneous arrival rates.

	Visiting order	$E[W_{total}]$
Proposal	1-2-7-3-4-5-8-6-1-2-9-3-4-5-10-6-	44,325.52±56.87
Cyclic	1-2-3-4-5-6-7-8-9-10-	49,357.16±49.41
	10-9-8-7-6-5-4-3-2-1-	49,416.77±52.19
TSP	sink-9-7-2-10-6-5-4-3-8-1-sink-	45,842.34±55.98
	sink-1-8-3-4-5-6-10-2-7-9-sink-	45,896.17±78.30

Table 3.6: Mean delivery delay $\mathrm{E}[W_{\mathrm{total}}]$ in the random layout model with heterogeneous arrival rates.

	λ_i	Visiting order	$\mathrm{E}[W_{\mathrm{total}}]$
Proposal	Descend.	1-2-3-7-1-4-5-1-2-8-6- 1-3-9-1-2-4-5-1-10-6-	31,749.81±39.45
	Random	3-2-1-4-3-5-7-6-8-3-2-4-9-3-5-6-10-	33,748.28±45.10
	Ascend.	10-1-2-3-4-5-10-6-7-8-9-	41,672.71±70.32
Cyclic	Descend.	1-2-3-4-5-6-7-8-9-10-	45,954.46±92.14
		10-9-8-7-6-5-4-3-2-1-	45,922.72±53.58
	Random	1-2-3-4-5-6-7-8-9-10-	46,237.65±48.98
		10-9-8-7-6-5-4-3-2-1-	46,478.53±87.18
	Ascend.	1-2-3-4-5-6-7-8-9-10-	46,961.42±37.85
		10-9-8-7-6-5-4-3-2-1-	46,982.49±34.47
TSP	Descend.	sink-9-7-2-10-6-5-4-3-8-1-sink-	45,053.78±69.97
		sink-1-8-3-4-5-6-10-2-7-9-sink-	47,838.46±77.27
	Random	sink-9-7-2-10-6-5-4-3-8-1-sink-	45,176.45±78.48
		sink-1-8-3-4-5-6-10-2-7-9-sink-	47,615.67±32.09
	Ascend.	sink-9-7-2-10-6-5-4-3-8-1-sink-	46,776.59±83.44
		sink-1-8-3-4-5-6-10-2-7-9-sink-	45,298.15±32.95

identical. Recall that cluster indices are set in the ascending order of S_i ($i \in \mathcal{N}$). Table 3.5 shows the result for the homogeneous arrival rate scenario. Our proposed scheme is superior to the TSP-based routing, which indicates that serving clusters close to the sink node more frequently is beneficial to the reduction of the overall mean delivery delay.

Finally, Tables 3.6 shows the results when both λ_i and S_i are heterogeneous. In all arrival rate scenarios, our proposed scheme shows the better performance than the TSP-based routing, and the difference between the mean delivery delays in our proposed scheme and the TSP-based routing depends on the scenarios. In general, a large variation in $\sqrt{\lambda_i/S_i}$ ($i \in \mathcal{N}$) yields the large variance of p_i ($i \in \mathcal{N}$), and it leads to a long visiting order sequence. Performance of our scheme has a strong correlation to the length of the visiting order sequence and scenarios yielding long sequences are more preferable for our proposed scheme.

3.6 Conclusion

We focused on a system where a message ferry collects bundles from isolated clusters and delivers those to the sink node, where transmission times of bundles are not negligible. To minimize the total mean delivery delay of bundles, we proposed an algorithm for obtaining a quasi-optimal visiting order of clusters, with the help of the optimization technique of the conventional polling model. Through simulation

experiments, we showed that the proposed visiting order can perform well, especially when the arrival rate and/or distances are heterogeneous.

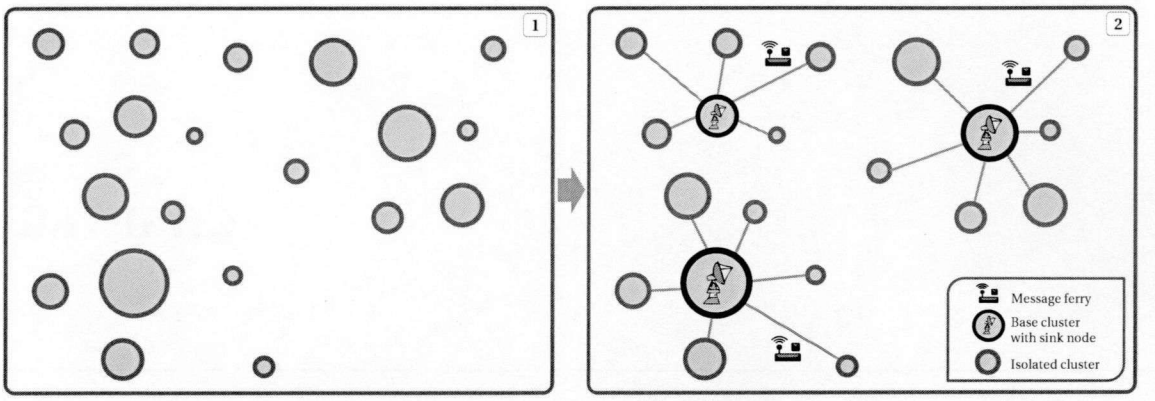
CHAPTER 4

Grouping Clusters

THIS chapter addresses the another part of inter-cluster communication by studying the grouping of clusters. Recall that when there are lots of clusters in ferry-assisted multi-cluster DTNs, multiple message ferries and sink nodes will be required to adopt with the system capacity limit. Hence, clusters need to be divided into groups such that each group consists of physically close clusters, a sink node, and a message ferry. In order to minimize the overall mean delivery delay of bundles, group should be created by taking account of the offered load of each cluster, distance between them and capacity limit of each sink node.

4.1 Grouping Clusters to Minimize the Total Mean Delivery

In the proposed multi-cluster delay tolerant networks, if the arrival rates of bundles at clusters are different from each other and service times are not negligible, bundles in clusters with high arrival rate must wait for a long time to be delivered to the sink node, while less important visits to clusters with a few bundles also take place. In Chapter 3 , we have already proposed and discussed a scheme to determine an optimal visiting order of a message ferry for one group [24], which minimizes the *mean delivery delay* of bundles, i.e., the average time interval from the generation of a bundle in a cluster to the completion of its delivery to the sink node. This optimization problem can be reduced to the minimization problem of the weighted mean waiting time in the conventional polling model of queueing theory [31, 53]. The proposed visiting order is effective, especially when arrival rates of bundles in clusters and/or



Scattered multi-cluster scenario .

Grouping clusters each consists of a base cluster with a sink node and a message ferry. Line implies connected group member with the base cluster.

Figure 4.1: Example of grouping where the size of each cluster is proportional to the arrival rate of bundles.

distances between clusters and the sink node are heterogeneous.

When there are lots of distant static clusters with heterogeneous offered load, there is potentially a drawback in designing a route using only one message ferry: The time spent for one cycle of the route increases with the number of clusters. This issue is the main concern of this chapter. The whole system is divided into multiple groups, each of which consists of a sink node, clusters, and one message ferry. We assume that the sink node is constructed in one of clusters in each group. In what follows, we call the cluster with the sink node the *base cluster* and others group members. Figure 4.1 presents an example of grouping. We further assume that high speed channels are available at the base cluster, so that the offered load of the base cluster is assumed to be excluded from the total offered load in each group. Note here that the total offered load handled by a message ferry should be less than one, and a moderate intensity, say 0.7 or less, is preferable. Moreover, for given number and positions of clusters, the total number of sink nodes (i.e., message ferries and groups) should be limited in order to suppress the introduction cost of the system. Figure 4.2 illustrates an example of grouping considering traffic intensity.

Our main goal is making groups to minimize the mean delivery delay of bundles among groups. As mentioned above, in Chapter 3, we have already obtained the solution in the case of one group [24]: We have the explicit objective function that in a nonlinear function composed of arrival rate of bundles in clusters and distance between clusters and their sink nodes. Based on this knowledge, we first model our problem as a nonlinear integer programming. Due to the complexity of the objective function, however, it might be hard to solve this problem directly. Furthermore,

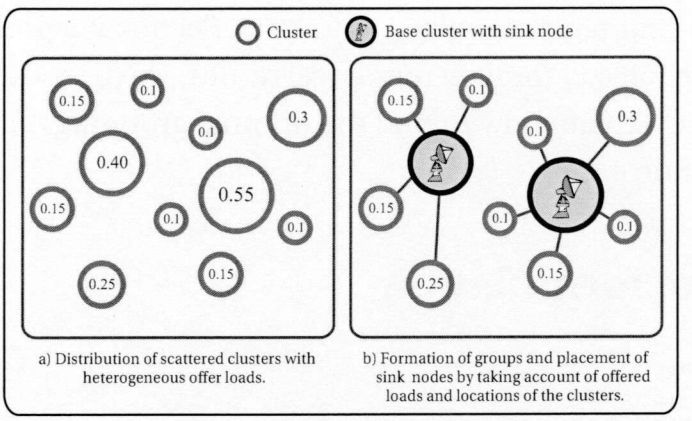


Figure 4.2: Example of grouping by considering traffic intensity ≤ 0.7 (the size of each cluster is proportional to the arrival rate of bundles and the numerical values inside the circle imply the average offered load of each cluster).

this formulation may sacrifice the performance of lightly-loaded clusters with long distances from their base clusters, in order to minimize the overall mean delivery delay.

To tackle these problems, we introduce two-step optimization technique based on linear integer programming. In the first step, we find the minimum of longest distances between group members and their base clusters under the constraint that the offered load in each group is less than a predefined threshold (e.g., 0.7). For this purpose, we use a variant of the capacitated vertex p-center problem (CVPCP) in facility location problems [33, 40, 41, 46].

Proposed Method: ***Capacitated vertex p-center problem:*** Capacitated vertex p-center problem tries to find locations of p capacitated facilities and assign customers to them when the locations and capacity of facilities, and the locations and demand of customers are given. The objective function of this problem is the minimization of the longest distance between customers and their associated facilities.

The first step optimization contributes to balancing the longest distance between a base cluster and its group members among groups. The second optimization reconfigures the groups in order to minimize the overall mean delivery delay. Because the objective function in the original problem is an increasing function of the square root of the product of group member's arrival rates and distances from their base clusters, we consider minimizing the sum of those products under the constraint that the longest distance does not exceed the first step optimization result. We give some numerical results to evaluate the characteristics of the obtained groups and how to find the optimal solution.

The rest of this chapter is organized as follows. Section 4.2 provides the problem formulation. In Section 4.3, through numerical results, we demonstrate the characteristics of groups and explain how to find the optimal grouping. Finally we conclude the chapter in Section 4.4.

4.2 Problem formulation

4.2.1 Overview

Our goal is the development of a method for dividing clusters into several disjoint groups adequately in terms of the introduction cost and the total mean delivery delay. For each group, we select a base cluster, where a sink node is located, and we assign a message ferry. Recall that the sink node has a connection to the outer world and can directly handle the traffic generated in its base cluster via high speed channels, and the message ferry goes back and forth between the base cluster and other clusters in order to collect bundles. The optimal visiting order of clusters in a group, which results in the minimization of the total mean delivery delay is obtained according to Chapter 3.

In general, the total mean delivery delay of bundles decreases with the increase of the number of message ferries (which is equal to the number of groups). Therefore our problem is multi-objective. In order to restrain the introduction cost, it is preferable that the number of message ferries should be minimal within a range that the total mean delivery delay is allowable. Note that the mean delivery delay in each group of clusters has the following two features, because each group of clusters can be viewed as a polling model [24] (discussed in Chapter 3).

1) The total offered load ρ handled by the message ferry should be moderate (e.g., $\rho \leq 0.7$) because the mean delivery delay is a nonlinear function of the total offered load ρ, which involves the factor $(1-\rho)^{-1}$.

2) Travel times between base cluster and group members linearly affect the mean delivery delay because they correspond to switchover times in the polling model.

Based on the above observation, we take the following approach. We first set the maximum allowable θ of offered load in each group and determine the lower bound K^*_{lower} of the number K of groups. We then attempt to solve a min-max integer program in order to minimize the total mean delivery delay of bundles. Note here that the number K of groups is first set to be K^*_{lower}, and if the program is not feasible,

we add one to K and solve the program again. Repeating this procedure, we will have the solution with a minimum feasible $K = K^*_{\min}$ eventually.

4.2.2 Nonlinear integer programming formulation

We assume that there are V clusters labeled 1 to V in a certain geographical area. Let $\mathcal{V} = \{1, 2, \ldots, V\}$ denote the set of cluster indices. We define $\boldsymbol{d} = [d_{i,j}]$ $(i, j \in \mathcal{V})$ as a matrix of the message ferry's traveling time $d_{i,j}$ between cluster i and cluster j, where $d_{i,i}$ $(i \in \mathcal{V})$ is equal to zero. Also, let $\boldsymbol{\rho} = [\rho_i]$ $(i \in \mathcal{V})$ denote a vector of the offered load ρ_i of cluster i. We assume that transmission times of bundles at all clusters are independent and identically distributed (i.i.d.) according to a general distribution with mean h_i. Let λ_i $(i \in \mathcal{V})$ denote arrival rate of bundles of cluster i.

We first find the lower bound K^*_{lower} of the number K of groups. Suppose there exist K disjoint, non-empty group partitions for a given maximum allowable offered load θ in each group. Without loss of generality, we assume $\rho_1 \geq \rho_2 \geq \cdots \geq \rho_V$. If cluster i for some $i > K$ is a base cluster of group k, there exists a cluster j $(j \leq K)$ of group k'. We then swap those two clusters; cluster j becomes a base cluster of group k and cluster i joins group k' as a group member. This swap yields another feasible group partition because it decreases the total offered load of group k' by $\rho_j - \rho_i$, and when $k \neq k'$, the total offered load of group k remains the same. Therefore, we consider only the case that clusters 1 to K are base clusters in discussing K^*_{lower} for a while. If a feasible partition of K groups is given, $\rho_{K+1} \leq \theta$ and $\rho_{K+1} + \rho_{K+2} + \cdots + \rho_V \leq K\theta$. We thus have

$$K^*_{\text{lower}} = \min_{1 \leq K \leq V} \{K;\ \rho_{K+1} \leq \theta,\ \sum_{i=K+1}^{V} \rho_i \leq K\theta\}.$$

Note that the minimum feasible number $K^*_{\min}$ of groups is not less than K^*_{lower}, i.e., $K^*_{\min} \geq K^*_{\text{lower}}$.

We define the set of base clusters as $\mathcal{K}$, where $|\mathcal{K}| = K$. Let $\mathcal{V}^{(k)}$ denote the set of clusters in group k, where $|\mathcal{V}^{(k)}| = V^{(k)}$. Let $\mathrm{E}[W^{(k)}_{\text{total}}]$ denote the overall mean delivery delay of bundles of group k. $\mathrm{E}[W^{(k)}_{\text{total}}]$ is defined as follows:

$$\mathrm{E}[W^{(k)}_{\text{total}}] = \frac{\sum_{i \in \mathcal{V}^{(k)} - \{k\}} \lambda_i \mathrm{E}[W^{(k)}_{\text{deliver},i}]}{\sum_{i \in \mathcal{V}^{(k)} - \{k\}} \lambda_i} \quad (k \in \mathcal{K}),$$

where the delivery delay $\mathrm{E}[W^{(k)}_{\text{deliver},i}]$ is the average time interval from the generation of a bundle of cluster i $(i \in \mathcal{V}^{(k)} - \{k\},\ k \in \mathcal{K})$ to the completion of its delivery to the sink

node in the base cluster at group k. The overall average weighted sum of total mean delivery delay of bundles of all groups becomes

$$\mathrm{E}[W_{\text{total}}] = \frac{\sum_{k\in\mathcal{K}} \lambda_{\text{total}}^{(k)} \mathrm{E}[W_{\text{total}}^{(k)}]}{\sum_{k\in\mathcal{K}} \lambda_{\text{total}}^{(k)}},$$

where $\lambda_{\text{total}}^{(k)} = \sum_{i\in\mathcal{V}^{(k)}-\{k\}} \lambda_i$.

Our main objective is to create groups of clusters in order to minimize $\mathrm{E}[W_{\text{total}}]$. Recall that minimization of $\mathrm{E}[W_{\text{total}}^{(k)}]$ can be obtained by optimizing the visiting order of the message ferry in each group k. The optimal visiting order of the message ferry can be achieved by adopting the minimization problem of conventional polling model as described in our previous work [24] (discussed in Chapter 3). In [24], it is obtained that by ignoring constant factors and terms, the objective function of the minimization problem is reduced to

$$f^{(k)}(\boldsymbol{q}^{(k)}) = \sum_{i\in\mathcal{V}^{(k)}-\{k\}} \frac{\lambda_i}{q_i}, \tag{4.1}$$

where $\boldsymbol{q}^{(k)}$ is a vector of q_i ($i \in \mathcal{V}^{(k)} - \{k\}$), which is the mean number of visits at cluster i per unit time at group k ($k \in \mathcal{K}$), i.e.,

$$q_i = \frac{1-\rho_{\text{total}}^{(k)}}{\sum_{j\in\mathcal{V}^{(k)}-\{k\}} \sqrt{2\lambda_j d_{k,j}}} \cdot \sqrt{\frac{\lambda_i}{2d_{k,i}}} \quad (i \in \mathcal{V}^{(k)} - \{k\}, k \in \mathcal{K}),$$

where $\rho_{\text{total}}^{(k)} = \sum_{i\in\mathcal{V}^{(k)}-\{k\}} \rho_i$. Therefore $f^{(k)}(\boldsymbol{q}^{(k)})$ in Eq. (4.1) is rewritten to be

$$f^{(k)}(\boldsymbol{q}^{(k)}) = \frac{\left(\sum_{i\in\mathcal{V}^{(k)}-\{k\}} \sqrt{2\lambda_i d_{k,i}}\right)^2}{1-\rho_{\text{total}}^{(k)}}. \tag{4.2}$$

Based on the above discussion, in this chapter the objective function of the grouping problem can be reduced to the minimization of weighted average of $f^{(k)}(\boldsymbol{q}^{(k)})$

among all groups:

$$\text{A:} \quad \text{minimize} \quad \frac{\sum_{k \in \mathcal{K}} \lambda_{\text{total}}^{(k)} f^{(k)}(\boldsymbol{q}^{(k)})}{\sum_{k \in \mathcal{K}} \lambda_{\text{total}}^{(k)}} \tag{4.3}$$

$$\text{subject to} \quad x_{i,j} \in \{0,1\}, \quad \forall i,j \in \mathcal{V}, \tag{4.4}$$

$$\sum_{i \in \mathcal{V}} x_{i,j} = 1, \quad \forall j \in \mathcal{V}, \tag{4.5}$$

$$\sum_{i \in \mathcal{V}} x_{i,i} = K, \tag{4.6}$$

where $x_{i,j}$ $(i,j \in \mathcal{V})$ are decision variables such that

$$x_{i,j} = \begin{cases} 1, & \text{if } i = j \text{ and cluster } i \text{ is a base cluster,} \\ 1, & \text{if clusters } i \text{ and } j \text{ are in the same group and} \\ & \text{cluster } i \text{ is a base cluster,} \\ 0, & \text{otherwise.} \end{cases}$$

Constraints (4.4) and (4.5) ensure that cluster j is either a base cluster ($x_{j,j} = 1$) or a cluster member in the same group as base cluster i ($x_{i,j} = 1$ for $i \neq j$). Constraint (4.6) implies that there are K base clusters. Therefore $\mathcal{K}$ and $\mathcal{V}^{(k)}$ can be defined by $x_{i,j}$:

$$\mathcal{K} = \{i;\ x_{i,i} = 1\}, \qquad \mathcal{V}^{(k)} = \{j;\ x_{k,j} = 1\} \quad (k \in \mathcal{K}).$$

As Eq. (4.3) is a nonlinear function, it might be hard to solve Problem A with a straightforward method. Furthermore, the mean delivery delay of lightly-loaded group members with long distances from their base clusters may get large because the minimization of the overall mean delivery delay will be achieved at the sacrifice of the bad performance of such clusters.

To tackle these problems, we take a two-step approach based on linear integer programming. From the original objective function Eq. (4.3), we expect that achieving the following two characteristics leads to our objective: a) Reducing and balancing total offered load among groups under certain capacity limitation, and b) reducing and balancing the total traveling distances among groups. In the next subsection, we show this can be realized by a two-step optimization technique based on linear integer programming. Note that total offered load among groups can be reduced by selecting clusters with high offered load as base clusters.

4.2.3 Linear integer programming formulation for approximate solution: Two-step optimization

From Eq. (4.2), we observe that the offered load $\rho_{\text{total}}^{(k)}$ in each group should be moderate, say, 0.7 or less. We then introduce the upper bound threshold θ of the offered load in each group. The first step in our approach is a relief of lightly loaded clusters. More specifically, given $\mathcal{V}$, K, $\boldsymbol{d}$, and θ, we first try to find grouping where the longest distance between base clusters and their group members is minimized under the constraint of K and θ. This can balance the longest distance among groups. Note here that this kind of problem can be best studied by the capacitated vertex p-center problem in facility location problems [40, 46]. Hence, we can formulate the first step optimization as the following modified version of the capacitated vertex p-center problem.

$$
\begin{aligned}
\text{B:}\quad & \text{minimize} && W \\
& \text{subject to} && x_{i,j} \in \{0,1\}, \quad \forall i,j \in \mathcal{V}, \\
& && \sum_{i\in\mathcal{V}} x_{i,j} = 1, \quad \forall j \in \mathcal{V}, \\
& && \sum_{i\in\mathcal{V}} x_{i,i} = K, \\
& && \sum_{j\in\mathcal{V}} \rho_j x_{i,j} - \rho_i x_{i,i} \le \theta x_{i,i}, \quad \forall i \in \mathcal{V}, \qquad (4.7) \\
& && \sum_{i\in\mathcal{V}} d_{i,j} x_{i,j} - W \le 0, \quad \forall j \in \mathcal{V}. \qquad (4.8)
\end{aligned}
$$

Constraint (4.7) implies that for a base cluster i, the total offered load in its group is not greater than θ. Note here that for $i \in \mathcal{V}$ such that $x_{i,i} = 0$, both left and right hand sides of constraint (4.7) are equal to zero. Constraint (4.8) ensures that the distance between a base cluster and group members in each group is not greater than W. Note that other constraints used in Problem B are the same as those used in Problem A. Recall that the initial value of K is set to be K_{lower}^* and is increased one by one to K_{min}^*, where a feasible solution is found. The solution gives us base clusters and a group partition of clusters, which minimize the maximum distance between base clusters and their group members.

By solving Problem B, we obtain the minimum $W = W^*$, which provides the maximum allowable distance between group members and their base clusters. Under this constraint, we then try to minimize the mean delivery delay of bundles. Unfortunately, however, the objective function of the original problem is nonlinear and it might be

difficult to solve it. We thus employ the following heuristics. From Eq. (4.2), we observe that the essential quantity in minimizing the mean delivery delay is $\sqrt{\lambda_i d_{k,i}}$ for group member i with base cluster k. Therefore, we reconfigure the groups to minimize the sum of $\sqrt{\lambda_i d_{k,i}}$ under the constraint of $\mathcal{K}$, θ, and W^*, where W^* is the solution of Problem B. The corresponding optimization problem is as follows.

$$
\begin{aligned}
\text{C:} \quad \text{minimize} \quad & \sum_{j \in \mathcal{V}} \sqrt{\lambda_j} \sum_{i \in \mathcal{V}} \sqrt{d_{i,j}} x_{i,j} \\
\text{subject to} \quad & x_{i,j} \in \{0,1\}, \quad \forall i,j \in \mathcal{V}, \\
& \sum_{i \in \mathcal{V}} x_{i,j} = 1, \quad \forall j \in \mathcal{V}, \\
& \sum_{i \in \mathcal{V}} x_{i,i} = K, \\
& \sum_{j \in \mathcal{V}} \rho_j x_{i,j} - \rho_i x_{i,i} \leq \theta x_{i,i}, \quad \forall i \in \mathcal{V}, \\
& \sum_{i \in \mathcal{V}} d_{i,j} x_{i,j} - W^* \leq 0, \quad \forall j \in \mathcal{V}.
\end{aligned}
$$

Note that in Problem C, the constraints are the same as those of Problem B except that $W = W^*$ is constant.

The remaining problem is finding optimal θ that satisfies Eq. (4.3). Given $x_{i,j}$ $(i,j \in \mathcal{V})$ by solving the two-step optimization problem, we can calculate Eq. (4.1) for each group. Therefore we can find the optimal θ as follows:

1. Set θ to be a maximum allowable offered load, e.g., 0.7.
2. Calculate the lower bound K^*_{lower} of the number K of groups according to the procedure in section 4.2.2.
3. Find the minimum feasible number K^*_{min} of groups according to the procedure in section 4.2.2.
4. With the help of line search technique [42], find the optimal $\theta = \theta^* \leq 0.7$, which minimizes the value of the objective function of Problem A. Note that the finally obtained grouping also minimizes $\mathrm{E}[W_{\text{total}}]$.

4.3 Numerical results

We consider an area of 40 [km] $\times$ 30 [km], where fifty isolated clusters ($V = 50$) are randomly located, as illustrated in Figure 4.3 and we then set $\boldsymbol{d} = [d_{ij}]$ $(i,j \in \mathcal{V})$

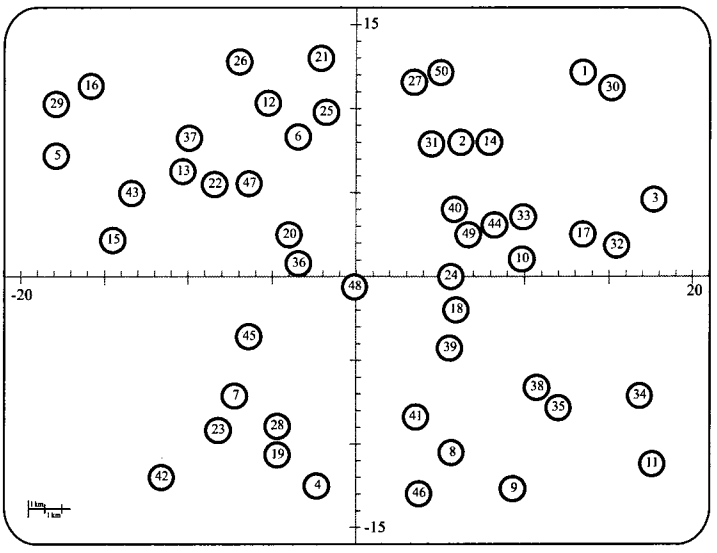


Figure 4.3: Random layout model with $V = 50$, where the numerical values inside circles imply the cluster IDs.

Table 4.1: Settings of ρ_i (V=50).

Case	ρ_1	ρ_2	ρ_3	$\cdots$	ρ_{50}	$\overline{\rho}$
Ascending	0.01	0.02	0.03	$\cdots$	0.50	0.255
Descending	0.50	0.49	0.48	$\cdots$	0.01	0.255

accordingly. For inter-cluster communications, we assume that each message ferry travels at a fixed speed of 10 m/s (i.e., 36 km/h). Table 4.1 illustrates two settings of $\boldsymbol{\rho}$ for heterogeneous and moderately loaded cases where ρ_i is assigned in an ascending order and a descending order with cluster IDs, and $\overline{\rho}$ is 0.255 in both cases. We assume that transmission times of bundles are i.i.d. according to an exponential distribution with mean $h_i = 1$ [s] ($i \in \mathcal{V}$). Since $\rho_i = \lambda_i h_i$ ($i \in \mathcal{V}$), the settings of λ_i ($i \in \mathcal{V}$) become identical to those of $\boldsymbol{\rho}$ in both cases. The total distance between base cluster k and its group members is denoted as $d^{(k)}_{\text{total}}$. By setting $\theta = 0.70$, we obtained $K^*_{\text{lower}} = K^*_{\min} = 12$ according to the procedures in section 4.2.2. Therefore we fix $K = 12$ in the rest of this section. We also found that the minimum feasible θ_{lower} of θ is given by 0.62, so that we consider $\theta \in [0.62, 0.70]$.

We obtain the groups by solving the two-step optimization technique using CPLEX [2]. Recall that Problem B provides temporary groups by minimizing the longest distance W between base clusters and their group members, while Problem C reconfigures the groups and provides final results by minimizing the sum of $\sqrt{\lambda_i d_{k,i}}$ under the constraint of the allowable longest distance W^*. Next, we determine the optimal visiting order of the message ferry in each group according to in Chapter 3 and [24]. Finally, we conduct the simulation experiments to obtain $\mathrm{E}[W^{(k)}_{\text{total}}]$ of group

Table 4.2: d_{total}, weighted average of $f^{(k)}(\boldsymbol{q}^{(k)})$, and $\mathrm{E}[W_{\text{total}}]$ ($K = 12$, Ascending case).

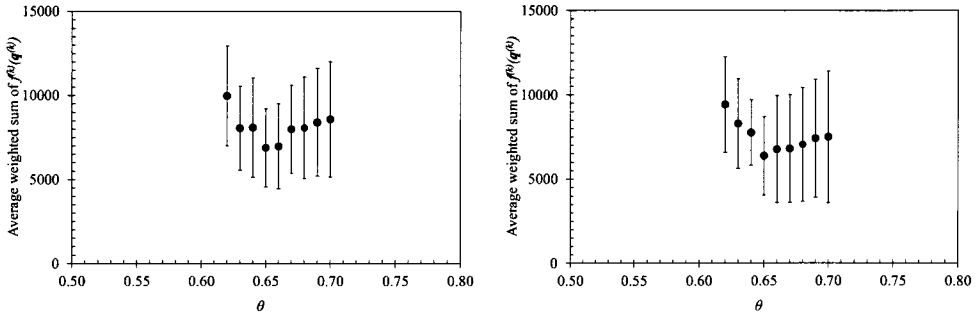
θ	d_{total} [km]		Weighted average of $f^{(k)}(\boldsymbol{q}^{(k)})$	$\mathrm{E}[W_{\text{total}}]$ [s]	
	Step 1	Step 2		Step 1	Step 2
0.70	300.1	272.6	8,277.5	7,088.7	6,481.6
0.69	302.8	278.1	8,345.1	6,891.8	6,434.5
0.68	308.9	279.5	8,011.2	6,714.0	6,411.2
0.67	313.5	281.4	7,815.3	6,610.1	6,302.1
0.66	321.7	284.2	6,925.2	6,419.4	5,581.6
0.65	331.2	285.9	6,890.0	5,496.2	5,061.5
0.64	349.5	330.0	8,092.8	6,608.1	6,480.6
0.63	353.8	331.8	8,056.1	6,645.8	6,487.0
0.62	417.1	416.2	9,977.0	8,326.2	8,198.9

Table 4.3: d_{total}, weighted average of $f^{(k)}(\boldsymbol{q}^{(k)})$, and $\mathrm{E}[W_{\text{total}}]$ ($K = 12$, Descending case).

θ	d_{total} [km]		Weighted average of $f^{(k)}(\boldsymbol{q}^{(k)})$	$\mathrm{E}[W_{\text{total}}]$ [s]	
	Step 1	Step 2		Step 1	Step 2
0.70	257.2	239.7	7,489.4	6,109.1	5,782.2
0.69	274.6	248.2	7,382.1	5,959.9	5,600.1
0.68	290.4	254.1	6,920.5	5,812.6	5,550.1
0.67	299.0	260.3	6,625.1	5,632.5	5,391.7
0.66	304.9	262.8	6,762.7	5,401.1	5,036.4
0.65	308.9	281.5	6,370.8	5,178.6	4,935.2
0.64	347.8	317.0	7,756.6	6,237.6	5,946.5
0.63	349.7	349.7	8,290.2	7,663.5	7,328.7
0.62	414.5	392.3	9,425.0	7,761.9	7,588.9

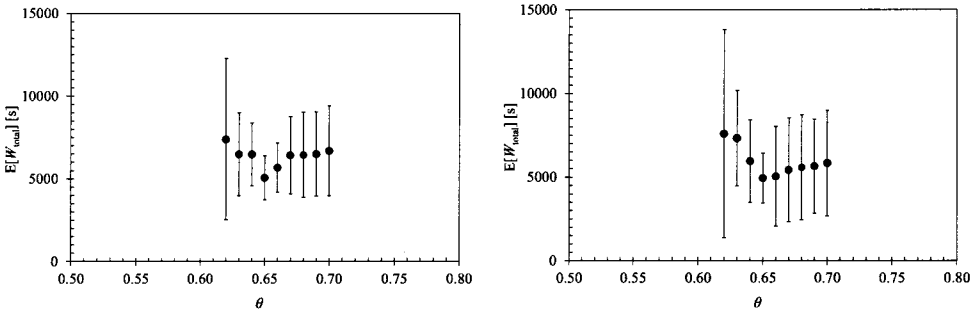
k and calculate $\mathrm{E}[W_{\text{total}}]$.

First, we observe the characteristics of grouping for different settings of θ in Table 4.2 for ascending case and in Table 4.3 for descending case. To grasp how the grouping of clusters changes, we show the sum d_{total} of distances $d_{k,i}$ of group members from their base clusters. As we expected, there is some room to improve the performance, regardless of θ, and $\mathrm{E}[W_{\text{total}}]$ decreases in Step 2. Next, when θ decreases, d_{total} monotonically increases while the weighted average of $f^{(k)}(\boldsymbol{q}^{(k)})$, which is the objective function in the original problem, initially decreases but increases from a certain value of θ. This suggests that there is an optimal $\theta^* = 0.65$. Figure 4.4 illustrates this characteristic of the weighted average of $f^{(k)}(\boldsymbol{q}^{(k)})$ for both cases. Note that error bars indicate the range of one standard deviation σ of the weighted average of $f^{(k)}(\boldsymbol{q}^{(k)})$ ($k \in \mathcal{K}$). We also observe that $\mathrm{E}[W_{\text{total}}]$ has the same tendency as the weighted average of $f^{(k)}(\boldsymbol{q}^{(k)})$ and the minimum $\mathrm{E}[W_{\text{total}}]$ is achieved at $\theta^* = 0.65$, which are shown in Figure 4.5 and Figure 4.6. Therefore, we can obtain the optimal θ by examining the weighted average of $f^{(k)}(\boldsymbol{q}^{(k)})$. Because of the limited search space for θ, this



(a) Ascending case. (b) Descending case.

Figure 4.4: Relationship between θ and weighted average of $f^{(k)}(\boldsymbol{q}^{(k)})$ ($K = 12$).

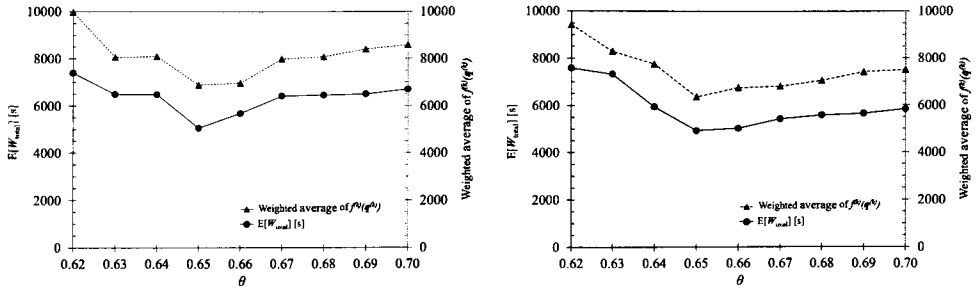


(a) Ascending case. (b) Descending case.

Figure 4.5: Relationship between θ and $\mathrm{E}[W_{\mathrm{total}}]$ ($K = 12$).

search does not require much computational overhead: θ should be not more than a moderate value, e.g., 0.7, and there will be the minimum feasible θ, θ_{lower}.

To examine the mean delivery delay in each group, we show $\rho_{\mathrm{total}}^{(k)}$, $d_{\mathrm{total}}^{(k)}$, and $\mathrm{E}[W_{\mathrm{total}}^{(k)}]$ in Table 4.4 ($\theta = 0.7$), Table 4.5 ($\theta = \theta^* = 0.65$), and Table 4.6 ($\theta = \theta_{\mathrm{lower}} = 0.62$) for ascending case. From Table 4.4, if $\theta > \theta^*$, clusters with lower offered load, e.g., 14 and 32, can become base clusters, which results in higher average of $\rho_{\mathrm{total}}^{(k)}$, i.e, 0.66. Note that the offered load of base cluster k is not included in $\rho_{\mathrm{total}}^{(k)}$. In addition, $\rho_{\mathrm{total}}^{(k)}$



(a) Ascending case. (b) Descending case.

Figure 4.6: Relationship between $\mathrm{E}[W_{\mathrm{total}}]$, weighted average of $f^{(k)}(\boldsymbol{q}^{(k)})$, and θ ($K = 12$).

Table 4.4: Results of two-step optimization ($K = 12, \theta = 0.7$, Ascending case).

Base cluster ID	14	32	37	38	39	42	43	45	46	47	49	50
$\rho_{\text{total}}^{(k)}$	0.64	0.64	0.69	0.69	0.68	0.70	0.69	0.62	0.63	0.69	0.68	0.58
$d_{\text{total}}^{(k)}$ [km]	22.4	17.3	32.7	30.3	21.4	22.9	24.6	19.2	25.8	23.1	13.4	19.5
$\mathrm{E}[W_{\text{total}}^{(k)}]$ [s]	3,373.1	2,890.5	7,462.7	6,424.4	4,525.1	5,769.2	5,855.5	3,665.7	3,898.2	4,960.5	2,574.5	1,689.6

Table 4.5: Results of two-step optimization ($K = 12, \theta = \theta^* = 0.65$, Ascending case).

Base cluster ID	33	35	37	39	42	43	44	46	47	48	49	50
$\rho_{\text{total}}^{(k)}$	0.64	0.64	0.64	0.64	0.58	0.65	0.63	0.64	0.64	0.65	0.64	0.63
$d_{\text{total}}^{(k)}$ [km]	34.9	35.2	20.9	18.8	24.4	28.2	17.7	26.6	15.7	15.9	8.5	39.1
$\mathrm{E}[W_{\text{total}}^{(k)}]$ [s]	5,373.8	4,568.3	3,036.2	2,414.4	3,091.9	4,272.5	3,015.2	3,385.0	2,353.2	2,929.5	1,001.2	3,128.1

Table 4.6: Results of two-step optimization ($K = 12, \theta = \theta_{\text{lower}} = 0.62$, Ascending case).

Base cluster ID	38	40	41	42	43	44	45	46	47	48	49	50
$\rho_{\text{total}}^{(k)}$	0.62	0.62	0.60	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62
$d_{\text{total}}^{(k)}$ [km]	32.2	21.4	27.1	36.6	37.3	23.9	42.6	34.6	22.1	57.2	38.7	42.5
$\mathrm{E}[W_{\text{total}}^{(k)}]$ [s]	4,755.9	2,928.3	4,658.6	4,867.4	5,605.6	3,859.9	6,093.8	5,523.8	3,289.3	8,596.9	4,938.5	5,717.9

is not well balanced: The maximum difference of $\rho_{\text{total}}^{(k)}$ among groups becomes 0.12. As a result, groups with high $\rho_{\text{total}}^{(k)}$ and large $d_{\text{total}}^{(k)}$ suffers high $\mathrm{E}[W_{\text{total}}^{(k)}]$, e.g., groups 37 and 38. If $\theta = \theta^*$ (Table 4.5), the average and standard deviation of $\rho_{\text{total}}^{(k)}$ is improved, i.e., 0.64±0.02, with a small increase of d_{total}: $d_{\text{total}} = 272.6$ for $\theta = 0.7$ and $d_{\text{total}} = 285.9$ for $\theta = 0.65$. This can be achieved by selecting clusters with high $\rho_{\text{total}}^{(k)}$ in the dense region. If $\theta < \theta^*$ (Table 4.6), due to the severe bound of θ, clusters with high $\rho_{\text{total}}^{(k)}$ become base clusters regardless of their locations. As a result, $d_{\text{total}}^{(k)}$ steeply increases and thus $\mathrm{E}[W_{\text{total}}^{(k)}]$ becomes worse. We observe the similar characteristics of grouping for descending case in Table 4.7 ($\theta = 0.7$), Table 4.8 ($\theta = \theta^* = 0.65$), and Table 4.9 ($\theta = \theta_{\text{lower}} = 0.62$).

Finally, examples of different grouping of clusters for both cases obtained by the combined scheme are illustrated in Figure 4.7 for different settings of θ. We find that lower $\mathrm{E}[W_{\text{total}}]$ can be achieved by two reasons: 1) Selecting clusters with high offered load as base clusters, and 2) balancing total offered load among groups. Note that the offered load of base cluster is not included in the total offered load $\rho_{\text{total}}^{(k)}$ of group k ($k \in \mathcal{K}$).

Table 4.7: Results of two-step optimization ($K = 12, \theta = 0.7$, Descending case).

Base cluster ID	1	2	3	4	5	6	7	8	11	12	13	24
$\rho_{\text{total}}^{(k)}$	0.58	0.7	0.53	0.42	0.66	0.7	0.66	0.67	0.59	0.69	0.7	0.69
$d_{\text{total}}^{(k)}$ [km]	11.5	28.7	10.8	13.7	18.4	20	20.7	32.9	17	16.2	24.8	28.6
$\mathrm{E}[W_{\text{total}}^{(k)}]$ [s]	1547.6	5287.1	1205.5	772.4	3285	4176.7	3252.8	6502.3	2443.9	3402.9	5504.7	4830.4

Table 4.8: Results of two-step optimization ($K = 12, \theta = \theta^* = 0.65$, Descending case).

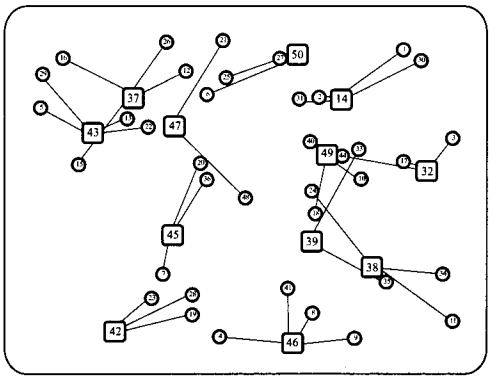
Base cluster ID	1	2	3	4	5	6	7	8	9	10	12	13
$\rho_{\text{total}}^{(k)}$	0.58	0.64	0.53	0.65	0.65	0.64	0.58	0.62	0.62	0.64	0.63	0.65
$d_{\text{total}}^{(k)}$ [km]	11.5	33.1	10.8	19.3	16.5	29	28.6	28.7	31	21.3	22.8	10.2
$\mathrm{E}[W_{\text{total}}^{(k)}]$ [s]	1547.6	4495.2	1205.5	3025.9	2755	3452	3246.5	4593	4619.3	3451.9	3651.8	1823.3

Table 4.9: Results of two-step optimization (K=12, $\theta = \theta_{\text{lower}}$=0.62, Descending case).

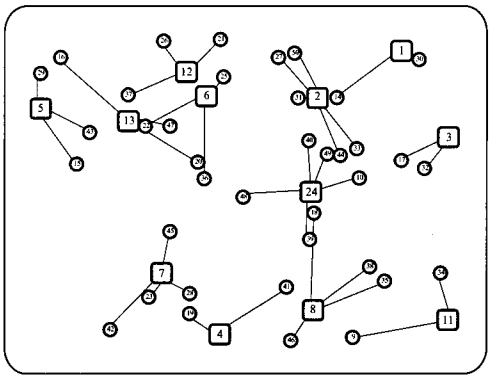
Base cluster ID	1	2	3	4	5	6	7	8	9	10	12	13
$\rho_{\text{total}}^{(k)}$	0.62	0.62	0.62	0.62	0.61	0.62	0.62	0.62	0.62	0.62	0.62	0.62
$d_{\text{total}}^{(k)}$ [km]	38.4	33.8	28.3	36.3	25.1	37.4	38.4	43	31	43.3	20.7	16.9
$\mathrm{E}[W_{\text{total}}^{(k)}]$ [s]	6244.5	5028.7	3770.5	5179.5	2558.3	5779.1	5013.9	5858.1	4619.3	6773.9	3222.1	2434.5

4.4 Conclusion

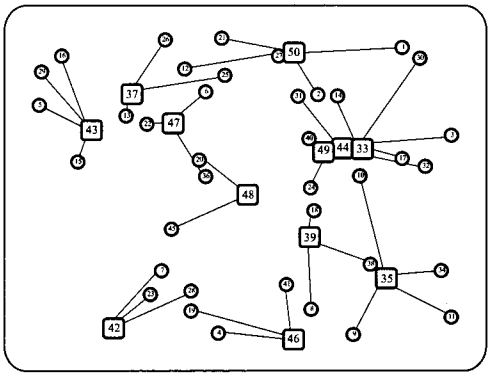
In this chapter, we focused on grouping clusters in ferry-assisted DTNs in order to minimize the mean delivery delay of bundles. We first modeled our problem as a nonlinear integer programming for exact solution. Due to the complexity of this problem, we further introduce two-step optimization technique based on linear integer programming for approximate solution. Through numerical results, we showed the two-step optimization can obtain optimal solution by setting θ adequately, which is realized using the original objective function as the stopping criterion.



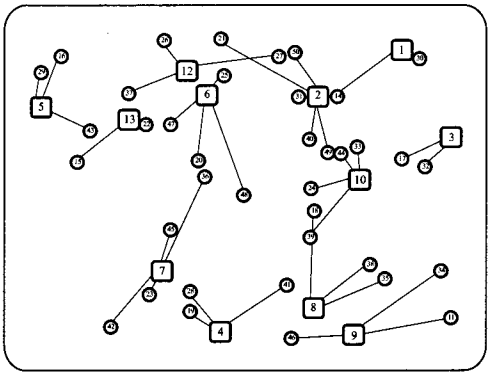
(a) $K = 12, \theta = 0.7$ (Ascending case).



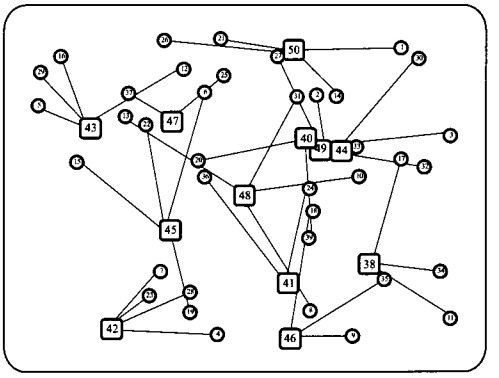
(b) $K = 12, \theta = 0.7$ (Descending case).



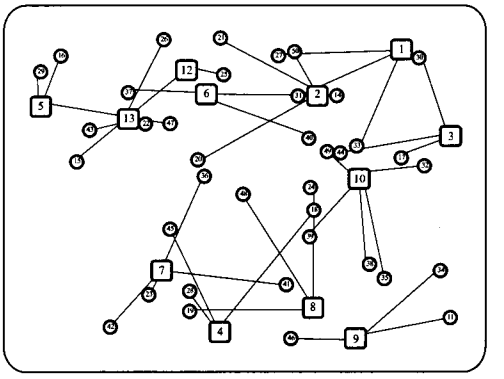
(c) $K = 12, \theta = \theta^* = 0.65$ (optimal ascending case).



(d) $K = 12, \theta = \theta^* = 0.65$ (optimal descending case).



(e) K=12, $\theta = \theta_{\text{lower}}$=0.62 (lower bound ascending case).



(f) K=12, $\theta = \theta_{\text{lower}}$=0.62 (lower bound descending case).

Figure 4.7: Grouping obtained by combined scheme for Fig 4.3 and Table 4.1, where rectangles are base clusters, i.e., locations of sink nodes and lines are drawn between sink nodes and their group members.

CHAPTER 5

Conclusions

IN this thesis, we proposed three techniques and studies to accomplish a complete system which can achieved efficient bundle gathering in ferry-assisted multi-cluster DTNs. We explicitly studied inter-cluster communication and intra-cluster communication carefully in order to minimize the total mean delivery delay of bundles. The inter-cluster communication was studied by two studies: 1) Grouping clusters, and 2) Optimal visiting order of isolated clusters. And, the intra-cluster communication was studied by self-organized data aggregation technique among selfish nodes in an isolated cluster. Figure 5.1 presents the summary of our proposed researches.

Chapter 2 addressed intra-cluster communication by self-organized data aggregation technique among selfish nodes in an isolated cluster. We proposed a self-organized data aggregation technique for collecting data from nodes efficiently, which can automatically accumulate data from nodes in a cluster to a limited number of nodes (called aggregators) in the cluster. 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. We further examined the proposed system in terms of successful data transmission, system survivability and the optimality of aggregator selection. We introduced two game models by taking account of the retransmissions mechanism of bundles. Through both theoretical and simulation-based approaches, we revealed feasible parameter settings that can achieve a system with desirable characteristics of stability, survivability, and successful data transfer.

Chapter 3 focused on one part of inter-cluster communication by studying the

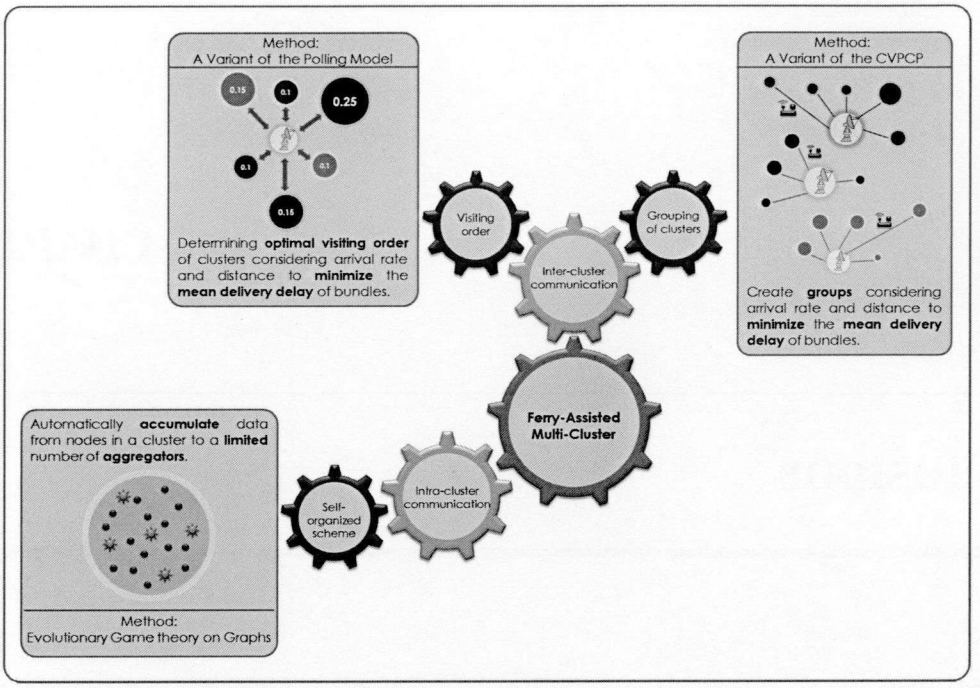


Figure 5.1: Proposed research studies.

optimal visiting order of isolated clusters. When there are lots of distant static clusters, the message ferry should visit them efficiently to minimize the mean delivery delay of bundles, where transmission times of bundles are not negligible. We proposed an algorithm for determining the optimal visiting order of isolated static clusters in ferry-assisted multi-cluster DTNs. We showed that the minimization problem of the overall mean delivery delay in our system is reduced to that of the weighted mean waiting time in the conventional polling model. We then solved the problem with the help of an existing approach to the polling model and obtain a quasi-optimal balanced sequence representing the visiting order. Through numerical examples, we showed that the proposed visiting order is effective when arrival rates at clusters and/or distances between clusters and the sink are heterogeneous.

Chapter 4 focused on another part of the inter-cluster communication by studying the grouping of clusters. When there are lots of distant static clusters, multiple message ferries and sink nodes will be required. We aimed to make groups each of which consists of physically close clusters, a sink node, and a message ferry, in order to minimize the overall mean delivery delay of bundles in consideration of both offered load of clusters and distance between clusters and their sink nodes. We first modeled this problem as a nonlinear integer programming, based on the knowledge obtained in our previous work [24] (discussed in Chapter 3). Because it might be hard to solve

this problem directly, we took two-step optimization approach based on linear integer programming, which yielded an approximate solution of the problem. Through simulation experiments, we showed that the optimal solution can be obtained by setting the control parameter appropriately, at least when clusters are distributed randomly over the area. Finally,a general guideline is proposed to achieve optimal grouping in ferry-assisted multi-cluster DTNs.

We hope that our proposed three techniques and studies to achieve a complete system for efficient bundle gathering in ferry-assisted multi-cluster DTNs can contribute to the delay tolerant networks (DTNs) communities.

Appendix A

Replicator Equation in Evolutionary Game Theory

The replicator equation [22, 50, 56, 59] is one of the fundamental equations in evolutionary game theory. Evolutionary game theory assumes that the population of a group (e.g., species) is proportional to the fitness (i.e., payoff) of the strategy that the group selects. Since each group is under mutual dependency with other groups, the superiority of the strategy is determined relatively by the strategy distribution.

We first formalize the general case for two players with n strategies. An $n \times n$ payoff matrix, $A = [a_{ij}]$, represents all possible strategy pairs of the two players. The entries, a_{ij}, $(i, j = 1, 2, \ldots, n)$, denote the payoff of strategy i competing with strategy j.

Let x_i $(i = 1, 2, \ldots, n)$ denote the ratio (relative frequency) of each strategy. All x_i add up to 1, i.e.,

$$\sum_{i=1}^{n} x_i = 1. \tag{A.1}$$

The expected payoff f_i of strategy i is given by

$$f_i = \sum_{j=1}^{n} x_j a_{ij}. \tag{A.2}$$

We can obtain the average payoff of the population to be

$$\phi = \sum_{i=1}^{n} x_i f_i. \tag{A.3}$$

It then follows from Eqs. (A.2) and (A.3) that the standard replicator equation is given as

$$\dot{x}_i = x_i(f_i - \phi), \quad i = 1, \ldots, n, \tag{A.4}$$

where a dot represents time derivative. Eq. (A.4) indicates that the number of players selecting strategy i increases with the relative difference between the expected payoff of strategy i and the average payoff of all strategies. Note that Eq. (A.4) is applicable only to an infinitely large and well-mixed population where each player can equally

play games with all other nodes [56].

Evolutionary game theory on graphs [32,35,37–39,51] is an extension of the original theory to a finite size population. Members of a population are represented by vertices of a graph and interact with connected individuals. It describes how the expected frequency of each strategy in a game changes over time within the graphs. The pair approximation [36] is applied to regular graphs of degree $k > 2$, i.e., each individual is connected to k other individuals. Each node represents a player with a selected strategy. Each player derives a payoff from interactions with all of its neighbors. Then, it compares the obtained payoff with a randomly chosen neighbor. If it overcomes the opponent, it keeps the current strategy, and otherwise, it imitates the strategy of the opponent. This kind of strategy updating rule is called "imitation updating rule."

By modifying the original payoff matrix A, the evolutionary game dynamics in a well-mixed population can be transformed into that on a k-regular graph. The modified payoff matrix, $A' = [a'_{ij}]$, is defined by the sum of the original $n \times n$ payoff matrix, $A = [a_{ij}]$, and an $n \times n$ modifier matrix, $M = [m_{ij}]$, where, m_{ij} describes the local competition between strategies i and j [37]. The transformed entries a'_{ij} of the modified payoff matrix, A' becomes

$$a'_{ij} = a_{ij} + m_{ij}.$$

In [37], m_{ij} for the imitation updating rule is defined as for $k > 2$,

$$m_{ij} = \frac{(k+3)a_{ii} + 3a_{ij} - 3a_{ji} - (k+3)a_{jj}}{(k+3)(k-2)}. \tag{A.5}$$

Note that off-diagonal elements of matrix M is anti symmetric, i.e., $m_{ij} = -m_{ji}$, because the gain of one strategy in local competition is the loss of another. Further, diagonal elements m_{ii} are always zero, suggesting that local competition between the same strategies results in zero. The expected payoff g_i for the local competition of strategy i is defined as

$$g_i = \sum_{j=1}^{n} x_j m_{ij}. \tag{A.6}$$

Note that the average payoff of the local competition of strategy i sums to zero, i.e.,

$$\sum_{i=1}^{n} x_i g_i = 0. \tag{A.7}$$

We thus obtain the average payoff ϕ of the population on graph to be

$$\phi = \sum_{i=1}^{n} x_i (f_i + g_i) = \sum_{i=1}^{n} x_i f_i, \tag{A.8}$$

which is the same as Eq. (A.3).

Let x_i denote the frequency of strategy i on a k-regular graph. Replicator equation on graphs can be obtained as follows [36–38]:

$$\dot{x}_i = x_i(f_i + g_i - \phi), \quad i = 1, \ldots, n, \tag{A.9}$$

where f_i, g_i, and ϕ are given in Eqs. (A.2), (A.6), and (A.8), respectively.

It is interesting to observe that Eq. (A.9) takes the same form as the standard replicator equation in Eq. (A.4), where the payoff matrix $[a_{ij}]$ is replaced by $[a_{ij} + m_{ij}]$. Therefore, many aspects of evolutionary dynamics on graphs can be analyzed by studying a standard replicator equation with the transformed payoff matrix $[a_{ij} + m_{ij}]$. Note that as k increases, the relative contribution of g_i decreases, compared to f_i, and in the limit of $k \to \infty$, Eq. (A.9) is reduced to Eq. (A.4). Therefore the replicator equation on a highly connected graph converges to the standard replicator equation [37].

References

[1] Delay tolerant networking research group (DTNRG). [Online]. Available: http://www.dtnrg.org

[2] IBM ILOG CPLEX Optimizer, version 12.2. [Online]. Available: http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/

[3] Interplanetary (ipn) internet project. [Online]. Available: http://www.ipnsig.org

[4] NetLogo, version 4.1.3. [Online]. Available: http://ccl.northwestern.edu/netlogo

[5] E. Altman, B. Gaujal, and A. Hordijk, "Balanced sequences and optimal routing," *Journal of the ACM*, vol. 47, no. 4, pp. 752–775, 2000.

[6] M. Ammar, D. Chakrabarty, A. Sarma, S. Kalyanasundaram, and R. Lipton, "Algorithms for message ferrying on mobile ad hoc networks," in *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, vol. 4, 2009.

[7] A. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[8] D. J. Bertsimas and X. Haiping, *Optimization of Polling Systems and Dynamic Vehicle Routing Problems on Networks.* Technical Report, OR 283-93, Massachusetts Institute of Technology, Operations Research Center, 1993. [Online]. Available: http://hdl.handle.net/1721.1/5356

[9] O. Boxma, W. Groenendijk, and J. Weststrate, "A pseudoconservation law for service systems with a polling table," *IEEE Transactions on Communications*, vol. 38, no. 10, pp. 1865–1870, 1990.

[10] O. Boxma, H. Levy, and J. Weststrate, "Efficient visit orders for polling systems," *Performance Evaluation*, vol. 18, no. 2, pp. 103–123, 1993.

[11] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay tolerant network architecture,," work in progress as an IETF RFC 4838 Draft. [Online]. Available: http://www.ietf.org/rfc/rfc4838.txt

[12] H. Chen and S. Megerian, "Cluster sizing and head selection for efficient data aggregation and routing in sensor networks," in *IEEE WCNC*, 2006, pp. 2318–2323.

[13] R. Conway, W. Maxwell, and L. Miller, *Theory of scheduling.* Dover Publications, 2003.

[14] P. Erdos and A. Renyi, "On the evolution of random graphs," *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, vol. 5, no. 17-61, p. 166, 1960.

[15] K. Fall, "A delay-tolerant network architecture for challenged internets," in *ACM SIGCOMM*, 2003.

[16] K. Fall and W. Hong, "Custody transfer for reliable delivery in delay tolerant networks," Intel Research Berkeley, Tech. Rep. IRB-TR-03-030, 2003.

[17] S. Farrell and V. Cahill, *Delay and Disruption Tolerant Networking.* Artech House, 2006.

[18] M. J. Ferguson and Y. J. Aminetzah, "Exact results for nonsymmetric token ring systems," *IEEE Transactions on Communications*, vol. COM–33, no. 3, pp. 223–231, 1985.

[19] C. Hauert and M. Doebeli, "Spatial structure often inhibits the evolution of cooperation in the snowdrift game," *Nature*, vol. 428, no. 6983, pp. 643–646, April 2004.

[20] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[21] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *International Conference on System Sciences*, 2000.

[22] J. Hofbauer and K. Sigmund, *Evolutionary Games and Population Dynamics.* Cambridge University Press, 1998.

[23] K. H. Kabir, M. Sasabe, and T. Takine, "Evolutionary game theoretic approach to self-organized data aggregation in delay tolerant networks," *IEICE Transactions on Communications,*, vol. E93-B, pp. 490–500, 2010.

[24] K. H. Kabir, M. Sasabe, and T. Takine, "Optimal visiting order of isolated clusters in dtns to minimize the total mean delivery time of bundles," *Numerical Algebra, Control and Optimization of American Institute of Mathematical Science (AIMS)*, vol. E93-B, pp. 490–500, December 2011.

[25] V. Kavitha and E. Altman, "Queuing in space: Design of message ferry routes in static ad hoc networks," in *21st International Teletraffic Congress.* IEEE, 2009.

[26] V. Kavitha and E. Altman, "Analysis and design of message ferry routes in sensor networks using polling models," in *8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt).* IEEE, 2010.

[27] V. Kavitha and E. Altman, "Opportunistic scheduling of a message ferry in sensor networks," in *2nd International Workshop on Mobile Opportunistic Networking.* ACM, 2010.

[28] J. Kurose and K. Ross, *Computer networking.* Pearson Addison Wesley, 2005.

[29] A. Kurs, A. Karalis, R. Moffatt, J. Joannopoulos, P. Fisher, and M. Soljacic, "Wireless power transfer via strongly coupled magnetic resonances," *Science*, vol. 317, no. 5834, pp. 83–86, 2007.

[30] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy-Kan, and D. B. Shmoys, *The traveling salesman problem: A guided tour of combinatorial optimization.* John Wiley & Sons, New York, 1985.

[31] H. Levy and M. Sidi, "Polling systems: Applications, modeling, and optimization," *IEEE Transactions on communications*, vol. 38, no. 10, pp. 1750–1760, 1990.

[32] E. Lieberman, C. Hauert, and M. Nowak, "Evolutionary dynamics on graphs," *Nature*, vol. 433, no. 7023, pp. 312–316, 2005.

[33] J. D. M. Albareda-Sambola and E. Fernández, "Lagrangean duals and exact solution to the capacitated p-center problem," *European Journal of Operational Research*, vol. 201, no. 1, pp. 71–81, 2010.

[34] H. Miura, D. Nishi, N. Matsuda, and H. Taki, "Message ferry route design based on clustering for sparse ad hoc networks," in *14th international conference on Knowledge-based and intelligent information and engineering systems*, 2010.

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

[36] H. Ohtsuki, C. Hauert, E. Lieberman, and M. Nowak, "A simple rule for the evolution of cooperation on graphs," *Nature*, vol. 441, no. 7092, pp. 502–505, 2006.

[37] H. Ohtsuki and M. A. Nowak, "The replicator equation on graphs," *Journal of Theoretical Biology*, vol. 243, no. 1, pp. 86–97, 2006.

[38] H. Ohtsuki and M. Nowak, "Evolutionary stability on graphs," *Journal of Theoretical Biology*, vol. 251, no. 4, pp. 698–707, April 2008.

[39] H. Ohtsuki, J. Pacheco, and M. Nowak, "Evolutionary graph theory: Breaking the symmetry between interaction and replacement," *Journal of Theoretical Biology*, vol. 246, no. 4, pp. 681–694, 2007.

[40] F. A. Özsoy and M. Ç. Pinar, "An exact algorithm for the capacitated vertex p-center problem," *Computers and Operations Research*, vol. 33, no. 5, pp. 1420–1436, 2006.

[41] S. P. M. S. R. Ahuja, J. Orlin and M. Scutellà, "A multi-exchange heuristic for the single-source capacitated facility location problem," *Management Science*, pp. 749–760, 2004.

[42] R. L. Rardin, *Optimization in Operations Research.* Prentice Hall, 1998.

[43] J. H. A. H. S. C. Borst, O. J. Boxma and G. B. Huitema, "Optimization of fixed time polling schemes," *Telecommunication Systems*, vol. 3, no. 1, pp. 31–59, 1994.

[44] L. Samuelson, *Evolutionary games and equilibrium selection.* MIT Press, 1997.

[45] S. Sano, N. Miyoshi, and R. Kataoka, "m-balanced words: A generalization of balanced words," *Theoretical computer science*, vol. 314, no. 1-2, pp. 97–120, 2004.

[46] M. Scaparra, S. Pallottino, and M. Scutellà, "Large-scale local search heuristics for the capacitated vertex p-center problem," *Networks*, vol. 43, no. 4, pp. 241–255, 2004.

[47] K. Scott and S. Burleigh, "Bundle protocol specification," work in progress as an IETF RCF 5050 internet draft. [Online]. Available: http://www.ietf.org/rfc/rfc5050.txt

[48] R. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: Modeling and analysis of a three-tier architecture for sparse sensor networks," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 215–233, 2003.

[49] K. Sigmund and M. A. Nowak, "Evolutionary game theory." *Current Biology*, vol. 9, no. 14, pp. 503–505, Jul 1999.

[50] J. Smith, *Evolution and the theory of games.* Cambridge University Press, 1982.

[51] G. Szabó and G. Fáth, "Evolutionary games on graphs," *Physics Reports*, vol. 446, no. 4–6, pp. 97–216, July 2007.

[52] H. Takagi, "Queuing analysis of polling models," *ACM Computing Surveys (CSUR)*, vol. 20, no. 1, pp. 5–28, 1988.

[53] H. Takagi, "Queueing analysis of polling models: an update," *Stochastic Analysis of Computer and Communication Systems*, 1990.

[54] A. S. Tanenbaum, *Computer Networks 4th Edition.* Prentice Hall, 2003.

[55] M. M. B. Tariq, M. Ammar, and E. Zegura, "Message ferry route design for sparse ad hoc networks with mobile nodes," in *ACM MobiHoc*, 2006.

[56] P. Taylor and L. Jonker, "Evolutionary stable strategies and game dynamics," *Mathematical Biosciences*, vol. 40, pp. 145–156, 1978.

[57] A. Traulsen, C. Hauert, H. De Silva, M. Nowak, and K. Sigmund, "Exploration dynamics in evolutionary games," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, no. 3, pp. 709–712, 2009.

[58] F. Warthman, "Delay-tolerant networks (dtns): A tutorial," 2003.

[59] J. Weibull, *Evolutionary game theory.* MIT Press, 1997.

[60] Z. Wu, X. Xu, and Y. Wang, "Does the scale-free topology favor the emergence of cooperation?" 2005. [Online]. Available: http://arxiv.org/abs/physics/0508220

[61] H. Yang and B. Sikdar, "Optimal cluster head selection in the leach architecture," in *IEEE International Performance, Computing, and Communications Conference*, 2007.

[62] O. Younis and S. Fahmy, "Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, pp. 366–379, 2004.

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

[64] W. Zhao and M. Ammar, "Message ferrying: Proactive routing in highly-partitioned wireless ad hoc networks," in *IEEE Future Trends of Distributed Computing Systems*. IEEE Computer Society, 2003.

[65] W. Zhao, M. Ammar, and E. Zegura, "Controlling the mobility of multiple data transport ferries in a delay-tolerant network," in *IEEE INFOCOM*, 2005, pp. 1407–1418.

Publications

A. Journal Papers

1. K. Habibul Kabir, Masahiro Sasabe, and Tetsuya Takine, "Evolutionary Game Theoretic Approach to Self-Organized Data Aggregation in Delay Tolerant Networks," *IEICE Transactions on Communications*, vol. E93-B, no. 03, pp. 490-500, March, 2010.

2. K. Habibul Kabir, Masahiro Sasabe, and Tetsuya Takine, "Optimal Visiting Order of Isolated Clusters in DTNs to Minimize the Total Mean Delivery Time of Bundles," *Journal of Numerical Algebra, Control and Optimization (NACO)*, vol. 1, no. 4, pp. 563-576, December, 2011.

B. Refereed Conference Papers

1. K. Habibul Kabir, Masahiro Sasabe, and Tetsuya Takine, "Design and Analysis of Self-Organized Data Aggregation Using Evolutionary Game Theory in Delay Tolerant Networks," in Proceedings of The Third IEEE WoWMoM Workshop on Autonomic and Opportunistic Communications, Kos, Greece, June, 2009 (6 pages).

2. K. Habibul Kabir, Masahiro Sasabe, and Tetsuya Takine, "Self-Organized Data Aggregation among Selfish Nodes in an Isolated Cluster," in Proceedings of the 1st Workshop on Bio-inspired Models and Technologies for Ambient Information Society (BioAmBIS 2010 workshop), Boston, MA, USA, December, 2010 (14 pages) (recipient of **"Best Workshop Paper Award"** in 5th International ICST Conference on Bio-Inspired Models of Network, Information, and Computing Systems).

3. K. Habibul Kabir, Masahiro Sasabe, and Tetsuya Takine, "Integer programming

formulation for grouping clusters in ferry-assisted DTNs," in Proceedings of the 2nd International Workshop on Complex Information Flows (CIF) of 26th IEEE International Conference on Advanced Information Networking and Applications (AINA), Fukuoka, Japan, March, 2012 (6 pages)(to be presented).

C. Domestic Conference

1. K. Habibul Kabir, M. Sasabe, and T. Takine, "Grouping clusters to minimize the total mean delivery delay of bundles in ferry-assisted DTNs," Technical Report of IEICE (NS2012-01), Okinawa, Japan, January, 2012 (6 pages)(to be presented).