MASTER OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING

Self-Organized Data Aggregation Techniques in Ferry Assisted Multi Cluster DTNs using a Novel Strategy of Evolutionary Game Theory

Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT)
Board Bazar, Gazipur-1704, Bangladesh.
October, 2015.

Self -Organized Data Aggregation Techniques in Ferry Assisted Multi Cluster DTNs using a Novel Strategy of Evolutionary Game Theory

by

Md. Salehin Ferdous Kader

MASTER OF SCIENCE

IN

ELECTRICAL AND ELECTRONIC ENGINEERING

Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT)
Board Bazar, Gazipur-1704, Bangladesh.
October, 2015.

CERTIFICATE OF APPROVAL

The thesis titled '**Self Organized Data Aggregation Techniques in Ferry Assisted Multi Cluster DTNs using a Novel Strategy of Evolutionary Game Theory**' submitted by Md. Salehin Ferdous Kader, St. No. 102610 of Academic Year 2010-11 has been found as satisfactory and accepted as partial fulfillment of the requirement for the Degree of MASTER OF SCIENCE in ELECTRICAL AND ELECTRONIC ENGINEERING on 15 October, 2015.

Board of Examiners:

1.

Dr. Khondokar Habibul Kabir (Supervisor) — Chairman
Assistant Professor,
Electrical and Electronic Engineering Department,
Islamic University of Technology (IUT), Gazipur.

2.

Dr. Md. Shahid Ullah (Ex-Officio) — Member
Professor and Head,
Electrical and Electronic Engineering Department,
Islamic University of Technology (IUT), Gazipur.

3.

Dr. Md. Ashraful Hoque — Member
Professor,
Electrical and Electronic Engineering Department,
Islamic University of Technology (IUT), Gazipur.

4.

Dr. Mohammed Imamul Hassan Bhuiyan — Member (External)
Professor,
Electrical and Electronic Engineering Department,
Bangladesh University of Engineering and Technology,
(BUET).

DECLARATION OF CANDIDATE

It is hereby declared that this thesis report or any part of it has not been submitted elsewhere for the award of any Degree or Diploma.

Dr. Khondokar Habibul Kabir (Supervisor)
Assistant Professor
Electrical and Electronic Engineering Department,
Islamic University of Technology (IUT), Gazipur.

Md. Salehin Ferdous Kader
Student No.: 102610
Academic Year: 2010-11
Date: 15 October, 2015.

Dedicated to my family and friends.

ACKNOLEDGEMENT

I would like to convey my heartiest gratitude to my supervisor Dr. Khondokar Habibul Kabir, Assistant Professor, Electrical and Electronic Engineering Department, Islamic University of Technology (IUT), Gazipur, for his encouragement and valuable advice provided to me during completion of my thesis. I couldn’t have completed my research without his continuous guidance and support.

I am indebted to Dr. Md. Shahid Ullah, Professor and Head, EEE Department, IUT for his valuable support throughout the two years of my M.Sc. course. I am grateful to Dr. Md. Ashraful Hoque, Professor, EEE Department, IUT for his priceless counsel on my academics.

I am thankful to Dr. Mohammad Imamul Hassan Bhuiyan, Professor, EEE Department, BUET for his precious advice on my research work.

I would like to specially thank to my colleague Mr. Muhammad Ahsan Zamee, Lecturer, EEE Department, Bangladesh University, for his valuable advice and continuous help regarding my thesis and also my friend Md. Mahmudul Hassan Nir for his continuous support and inspiration.

I would like to thank all the faculty members and staff of the department of EEE, IUT for their inspiration and help. I would like to thank my parents for their blessings and my family and friends for having faith in me throughout the times of this thesis.

Finally, I would like to express my unparalleled gratitude to Almighty Allah for his divine blessing without which it would not have been possible to complete this thesis successfully.

Md. Salehin Ferdous Kader
October, 2015

List of Abbreviation of Technical Terms

DTN	Delay Tolerant Network
GT	Game Theory
EGT	Evolutionary Game Theory
IM	Imitation
BD	Birth-Death
DB	Death-Birth
MANET	Mobile ad-hoc networks
UAV	Unmanned aerial vehicle
OPWP	Optimized Way-points
TCP	Transmission Control Protocol
DSL	Digital Subscriber Line
DARPA	Defense Advanced Research Projects Agency
HEED	Hybrid, energy-efficient, and distributed
ADU	Application-defined data unit
LEACH	Low-energy adaptive clustering hierarchy
ESS	Evolutionarily Stable Strategy
IP	Internet protocol
ESS	Evolutionary Stable Strategy
OPWP	Optimized Way Points
NASA	Defense Advanced Research Projects Agency

Abstract

Delay Tolerant Networks (DTN) can provide data connectivity for areas where internet cannot provide any end to end connectivity. In DTNs, store-carry-forward mechanism with the help of custody transfer technique provides reliable end-to-end data transfer where nodes which transfer data with custody called custodians. In this system, sometimes storage congestion may occur. This drawback can be improved by using special mobile nodes called message ferries. In such scenario, it is better to aggregate data to some custodians so that the message ferry can efficiently collect them and carries collected bundles to a base station referred to as a sink node. When there are several isolated clusters (formed by physically close wirelessly connected nodes), message ferry have to visit those clusters and collects bundles from custodians. When message ferry visits any individual cluster (referred to as intra-cluster visit or visit within cluster), sometimes it is not possible to visit and collect data from all nodes in that cluster within particular period of time. To resolve this problem, self-organized data aggregation technique is developed, where, with the help of the evolutionary game theoretic approach, the system can automatically select some limited number of nodes (custodians) called aggregators. As a result, the message ferry visits and collects bundles only from those aggregators in that particular cluster. In this technique, Aggregators are changed autonomously in each round. In self-organized data aggregation technique, the strategy (i.e., to become aggregator or sender) of selection of aggregators is modeled as a game in evolutionary game theory, where, each node will draw payoff after interaction. For strategy selection, Imitation Update Rule of evolutionary game theory was used previously. In this research, we compared the existing solution of Imitation Update rule with Birth-Death and Death-Birth Update Rules of evolutionary game theory and select the better update rules among them for selection of Strategy in Self-Organized Data Aggregation Technique for the cases of no retransmission and retransmission. For this, we proposed our mathematical model and after the numerical analysis, we have select Birth-Death Update Rule is as better update rule for the case of no retransmission and Death-Birth Update rule is as better update rule for the case of retransmission when degree is smaller and for higher degree, Imitation Update Rule as better update rule.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Background

As internet is the most vital element in today's life so we can't think today's life without internet. The people who live in cities get all the amenities of modern technologies. But unfortunately, the scenario is not quite the same in most of the developing countries or the third-world countries, as the rural people here are greatly deprived of ocean of information available in today's Internet. Whereas, immense use of the information technology could be a great catalyst for rural development. Again, the place which is very distant from advantages of the urban life, it is difficult to deliver and collect information from those areas. Even in infrastructure less areas and the areas which are in bad condition after some disasters, it is also very difficult to gather information from them as there is no end to end connectivity or global Internet connection.

In our case, we will take these types of infrastructure less or distant areas on account. In this chapter, a brief background and motivation of the research are discussed, which includes the internet, a brief about Delay Tolerant and some other wireless networks outside the internet.

1.1 Introduction

1.1.1 Background and present state of the problem

The Internet has been a great success at interconnecting communication devices across the Earth. It has done this by using a homogeneous set of communication protocols, called the *TCP/IP protocol suite*. All devices on the hundreds of thousands of networks that make up the Internet use these protocols for routing data and insuring the reliability of message exchanges. Connectivity on the Internet relies primarily on wired links, including the wired telephone network, although wireless technologies such as satellite and short range mobile links are also an essential part of the network. These links, as used on the Internet, are continuously connected in end-to-end, low-delay paths between sources and destinations. They have low error rates and relatively symmetric bidirectional data rates.

In ambient information society, it is expected that each user can automatically obtain its desired information from environments equipped with a numerous number of devices.

The under lying network supporting the ambient information society can be regarded as a kind of delay tolerant networks (DTNs) [1, 2] due to lack of reliable continuous end-to-end connectivity.

The Internet allows people to communicate from far distances. It is a great opportunity for many people and the economy. Nevertheless, not everyone has access to these technical facilities. Some areas, especially developing countries and rural areas do not have this chance, hence increasing the gap between developed and developing countries. In other situations, such as recently in the Arab world, access to Internet is disabled and prevents people from communicating with the rest of the world. By using, *delay tolerant network* (DTN) [7, 9, 11], data connectivity for these kinds of scenarios can be established by *store carry forward* mechanism.

DTN is an approach to computer network architecture that seeks to address the technical issues in heterogeneous networks that may lack continuous network connectivity. A DTN is a network of smaller networks. It is an overlay on top of special-purpose networks, including the Internet [16]. Examples of such networks are those operating in mobile or extreme terrestrial environments, or planned networks in space. Recently, the term disruption-tolerant networking has gained currency in the United States due to support from DARPA, which has funded many DTN projects [1]. Disruption may occur because of the limits of wireless radio range, sparsely of mobile nodes, energy resources, attack, and noise.

In (DTNs), *custody data transfer* [8] mechanism is used. It deals reliable end-to-end data transfer in which *custodians* transfer data with custody. Custodians are the intermediate nodes which keep custody bundles. The custodians should have a huge space to hold the data custody until there is a successful data transfer. So to be a custodian, a node must reserve a sufficient amount of storage. Also custodians generate their own data .So space congestion may occur due to huge data. Due to space congestion, they refuse to take custody of other nodes. As every custodians also generate its own data, energy of battery decrement is a problem. It is selfish behavior of nodes to save their own battery life.

To solve this congestion problem some special mobile nodes can be introduced to proactively travel the network and gather bundles from the custodians before congestion occur. This is called message ferries.

Message ferries [1-3] can solve the storage congestion problem by actively visiting the network and gather bundles from the custodians. Note that the message ferry has a sufficient amount of storages and energy to carry the bundles to the corresponding destination, i.e., a base station referred to as *sink node* [1-3], and it can also supply energy to the nodes if required. When there are several isolated networks referred to as *clusters,* the message ferry must visit each of the cluster and collects bundles from the custodians as shown in Fig.1.1. In such kind of scenarios, however, sometimes it is difficult for the message ferry to visit all of the nodes in a certain period of time. Taking account of the challenges, we developed a self-organized data aggregation technique in [4]. With the help of the evolutionary game theoretic approach [44, 45, 47], our system can automatically select some special custodians referred to as *aggregator* , which are cooperative in nature and willingly hold custody bundles of other nodes referred to as *senders*.

Therefore, the message ferry [15] needs to collect the bundles only from the aggregators. Note here that in this scheme, each aggregator must keep awake to receive and hold the bundles until transferring them to the message ferry, while each sender awakes only when generating and sending the bundles. In addition, each aggregator can obtain energy supply from the message ferry only when it finds a sender as its neighbor. In our scheme, each node appropriately selects strategy, i.e., sending or aggregating, depending on neighbors' strategies. This interaction among nodes is modeled as a game in game theory. As clusters are formed by physically close wirelessly connected nodes, sometimes it is not possible for the message ferry to visit all the nodes in a cluster and collects data individually in a certain period of time. Taking amount of these challenges on mind, *self-organized data aggregation technique* [1-4] is developed. With the help of the evolutionary game theoretic approach , the system can automatically select some custodians referred to as *aggregators*, which are cooperative in nature and willing to hold custody bundles of other nodes referred to as senders. Therefore, message ferries need to collect the bundles only from the aggregators.

Each node is equipped with a long range radio and a short range radio [1-4]. While the message ferry approaching to the cluster it broadcasts its availability to all members of the cluster. Only aggregators with specific amount of bundles are allowed to transmit service request to the message ferry by their long range radio. These service request message contains: 1) information of all aggregators location.2) the amount of bundles it wants to transmit. To guide the message ferry, the aggregator occasionally transmit location update message on reception of each information, the message ferry calculate the intra cluster path. When the message ferry and one aggregator is close enough, the aggregator transfer bundles by it short range radio to the message ferry .At the same time, it get energy supply from the message ferry.

Initially each node randomly chosen to be an aggregator or a sender because it can’t know the neighbors behavior. In the successfully rounds, each nodes selects their role depending on their results of the previous round with the help of Evolutionary Game theory. Due to lack of reliable connectivity among arbitrary nodes, it is difficult to achieve centralized controls in DTN.

Note here that in this scheme, each aggregator should keep awake to receive 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. If both nodes are selected to be an aggregator, they lose largest energy without any energy supply from the message ferry because they are not able to collect sufficient number of bundles to request the message ferry to visit. At the initial time none of the cluster member have any bundles. While some cluster member generate their own bundles, they seek for aggregators within the transmission range. If no aggregator is available, the initial bundle generators become aggregators.

The aggregators are changed in each round [1-4] (as shown in Fig.1.1) (The unit time at which the role of each node is changed) in decentralized way as per the requirement of energy [1-4]. The number of aggregator can be controlled by adjusting the energy that the message ferry supplies to the aggregators.

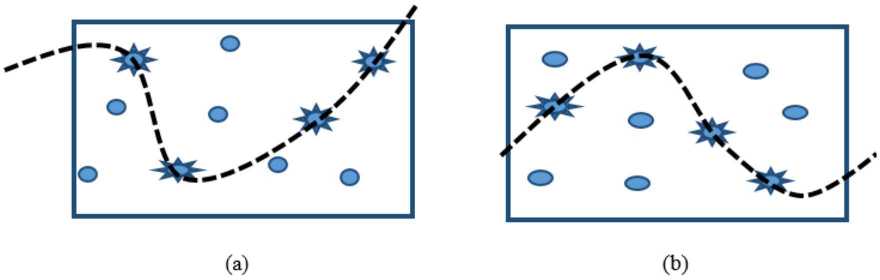


Fig.1. 1 Role of aggregators are changing after each round. In figure (b), the aggregators have changed their roles than from figure (a)

In self-organized data aggregation technique [1-4], the strategy (i.e., to become aggregator or sender) of selection of aggregators is modeled as a game in evolutionary game theory [9], where, each node will draw payoff after interaction.

For strategy selection, [1-4] used Strategy Imitation update rule of Evolutionary Game Theory [44, 47, 48]. Now, we will review some literature which works on Self-Organized Data Aggregation Techniques using Evolutionary Game Theory.

Evolutionary game theory (EGT) is the application of game theory to evolving populations of lifeforms in biology. We will discussed Self-Organized Data Aggregation Techniques [1-4] and Evolutionary Game Theory in details in Chapter 3 and Chapter 4. After reviewing the literature, we discuss about our proposed Solution to the problem

1.2 Literature Review

There is a lot of work going on about the related topics on Self-Organized Data Aggregation Techniques using Evolutionary Game Theory. In previous work, Aggregator selection is similar to cluster-head selection in clustering schemes [1-4]. 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) [1-4] and hybrid, energy-efficient, and distributed (HEED) [1-4] clustering approach are well-known schemes for wireless sensor networks.

LEACH [1-4] 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 [1-4] 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 [1-4] improves network lifetime over LEACH [1-4]. 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) [1-4], rather than the performance of the whole system (e.g., system lifetime). Evolutionary game theory [7, 9] 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.

Evolutionary game theory [44, 45] provides us with both theoretical framework and simulation-based framework. The theoretical framework called replicator dynamics [44,45,47] is a mathematical model, where the ratio of individuals selecting a strategy increases when the strategy can yield more payoff [1-4] than the average payoff of the whole system .The replicator dynamics [43,44,45] 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 .They derived replicator dynamics on graphs for three kinds of strategy-updating rules: Birth-death updating, death-birth updating, and imitation updating.

K. Kabir et. al., in 2011, proposed an algorithm for determining the optimal visiting order of isolated static clusters in DTNs [52]. In 2012, K. Kabir et. al. 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 data bundles [51]. They developed simple heuristic algorithms and through simulation experiments showed that the optimal solution can be obtained by appropriate parameter setting, when the clusters are randomly distributed over the area. In 2009, K. Kabir et. al. proposes a scheme to aggregate data into selected custodians, called aggregators, in a fully distributed and autonomous manner by using evolutionary game theoretical approach where they could also control the number of aggregators to a desired value [4]. In 2010, K. Kabir et. al. proposed a scheme to aggregate data into selected custodians, called aggregators, in a fully distributed and autonomous manner with the help of evolutionary game theoretic approach [3]. In 2010, they further examined the proposed system in terms of success of data transmission and system survivability [1]. In 2013, they even further examined the proposed system in terms of successful data transmission, system survivability and the optimality of aggregator selection [52]. They also first introduced a new game model with retransmissions.K. Habibul Kabir, Masahiro Sasabe, and Tetsuya Takine applied 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.

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

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, also consider a system with *mutation updating* [52].

After reviewing all the above literature, we find that to select the strategy, Imitation Update rule of Evolutionary Game Theory has been used in [1], [2] , [3] and [4]. However, to select the strategy, Strategy Birth-Death and Death-Birth Update rules are not used yet, our aim is to compare the existing Imitation updating rule with the Birth-Death and Death-Birth Updating rule and find the better updating rule among three updating rule

1.3 Our Proposal: Best Update Rule selection

In this research, our goal is to compare the existing method for strategy selection using Imitation Update rule [1-4] with Birth-Death and Death-Birth Update Rule of Evolutionary Game Theory [9] and select the Novel Strategy among them for Self-Organized Data Aggregation Technique [1-4]**.** We are selecting these two rules because they are the part of three update rules of Evolutionary game theory and for selecting the strategy of aggregators, Imitation Update Rule of Evolutionary Game Theory was used [1-4] but Birth-Death and Death-Birth Update Rules haven't used yet.

For doing this, we have to follow the steps below:

i. At first, develop the modifier matrix from payoff matrix for two different Strategy update rules i.e. *Birth-Death* and *Death-Birth Update rules* and compare them with existing Imitation *Update Rule.*
ii. Derive the replicator equation on graph for *Birth-Death* and *Death-Birth Update Rules*.
iii. Find the solution of *Birth-Death* and *Death-Birth Update Rules* with the help of the replicator equation on graph, and then find the suitable equilibrium conditions.
iv. Investigate the appropriate value of different related parameters (energy supplied by message ferry, *b*; energy consumed by sender, *s* and energy consumed by aggregator, *c*) to fulfill the equilibrium condition.
v. Compare the solutions and equilibrium of *Birth-Death* and *Death-Birth Update Rules* with exiting solution and equilibrium of *Imitation Update Rule* select the better update rule with appropriate conditions for selection of novel strategy

1.4 Thesis Organization:

This thesis will compare and select the best strategy for selection of aggregators in Ferry Assisted Multi Cluster DTNs. The thesis organized in following manner.

In **Chapter 2**, Ferry Assisted Multi Cluster DTN model will be discussed.
In **Chapter 3,** Self-Organized data Aggregation techniques will be discussed.
In **Chapter 4,** Game Theory, Evolutionary Game theory and Replicator Equation will be discussed.
In **Chapter 5,** the mathematical Model of our work will be discussed.
In **Chapter 6,** Numerical Analysis of given mathematical model will be discussed.
In **Chapter 7,** Conclusion of the thesis is discussed where recommendation for future work is given.

Chapter 2

Ferry Assisted Multi Cluster DTNs

It is known that DTN is a unique technique that enables communication and data transfer in situations where traditional networks are obsolete, i.e., where, physical end-to-end connectivity is not present [1, 2]. By using DTN technique, we can practically establish connectivity within a closed remote boundary using wireless equipped devices, where devices connect each other using Ad-Hoc networking.

This chapter discusses the Ferry-Assisted Multi Cluster DTN in details.

2.1 Delay Tolerant Network (DTN)

DTN is a message-based store-and-forward overlay network architecture. Unlike IP networks that are based on fixed-length packets, DTN operates on application-defined data units (ADUs) called Bundles. Each bundle contains arbitrary application content in its payload, along with addressing and an extensible set of other protocol blocks. Unlike most IP-based protocols in which metadata is stored in protocol headers, bundle metadata may appear either before or after the payload, hence the term "block" is used instead of header. These blocks contain the address and policy information used for routing, as well as information relating to reliability protocols and security management. The blocks are extensible, both for purposes within the infrastructure, as well as for applications to attach additional content.

DTN can leverage persistent storage resources within the network to buffer bundle data while it is in transit. This is unlike most typical routers which buffer data only in volatile memory while it is being processed. In the DTN environment, storage is used to wait for connectivity to be restored to some destination before transmitting a message, or to save the state of the system in case of a power outage. In part as a consequence of this buffering design, bundles have a real-time expiration lifetime parameter that is set when the bundle is generated and controls how long the bundle should remain in the network before it is either delivered or proactively deleted to reclaim resources.

This design allows the application to set a validity interval for transmitted messages, and is used as one way for the system to cope with the fact that it may take a potentially long amount of time for the message to reach its destination. This storage is also leveraged for DTN’s reliability mechanism called custody transfer. The idea of custody transfer is that responsibility for delivery of a bundle can be transferred between 30 nodes in the network (custodians), as the bundle proceeds along its path to the eventual destination. This means that once a node has accepted custody of a message, it has a responsibility to expend additional resources to both reliably store a copy of the bundle as well as to route the bundle to its destination, potentially requiring multiple transmissions, until either the bundle is delivered or some other node takes custody. In contrast, in the Internet architecture, responsibility for reliable delivery exists at the endpoints, due to the expectation that the network tends to be connected most of the time and that latencies are minimal. However, the expectation of intermittent outages and potentially long end-to-end latencies in DTN environments means that in many cases, the performance improvements gained by custody transfer justify its complexity (and thus falls within the suggested principals espoused by the “end-to-end argument”. Also in some cases, other nodes within the network are more capable of executing a reliable delivery than the originating source node, which may be constrained (or in the extreme case, about to fail permanently).

2.1.1 History of Delay Tolerant Network

Researchers began developing technology for routing between non-fixed locations of computers [7-9]. While the field of ad hoc routing was inactive throughout the 1980s, the widespread use of wireless protocols reinvigorated the field in the 1990s as mobile ad hoc networking (MANET) and vehicular ad hoc networking became areas of increasing interest. Actually it started in the end of 1970s, spurred by the decreasing size of computers. Concurrently with (but separate from) the MANET activities, DARPA had funded NASA, MITRE and others to develop a proposal for the Interplanetary Internet (IPN).

In 2002, Kevin Fall started to adapt some of the ideas in the IPN design to terrestrial networks and coined the term delay-tolerant networking and the DTN acronym. A paper published in 2003 SIGCOMM conference gives the motivation for DTNs [36]. The mid-2000s brought about increased interest in DTNs, including a growing number of academic conferences on delay and disruption-tolerant networking, and growing interest in combining work from sensor networks and MANETs with the work on DTN.

2.1.2 DTN uses

Many evolving and potential communication environments do not conform to the Internet's underlying assumptions. These environments are characterized by:

I. **Intermittent Connectivity**: The absence of an end-to-end path between source and destination is called network partitioning. In such cases, communication using the TCP/IP protocols does not work [17].

II. **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 acknowledgements or data [11].

III. **Asymmetric Data Rates:** The Internet supports moderate asymmetries of bidirectional data rate for users with cable TV or asymmetric DSL service. But if asymmetries are large, they defeat conversational protocols.

IV. **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 retransmission than for Internet-type end-to-end retransmission (linear increase vs. exponential increase, per hop).

V. **Intermittent connectivity:** A growing number of communicating devices are in motion and operate on limited power. This is true in interplanetary space and is becoming more common on Earth among mobile wireless communication devices, such as cell phones. When communicating nodes are in motion, links can be obstructed by intervening bodies. When nodes must conserve power or preserve secrecy, links are shut down. These events cause intermittent connectivity [17]. When no path exists to connect a source with a destination, a network partition is said to occur.

2.1.3 Store-and-Forward Message Switching

DTNs overcome the problems associated with intermittent connectivity, long or variable delay, asymmetric data rates, and high error rates by using *store-and-forward message switching*. This is a very old method, used by pony-express and postal systems since ancient times. 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. Store-and-forwarding methods are also used in today's voicemail and email systems, but these systems are not node-to-node relays (as shown above) but rather star relays; both the source and destination independently contact a central storage device at the center of the links.

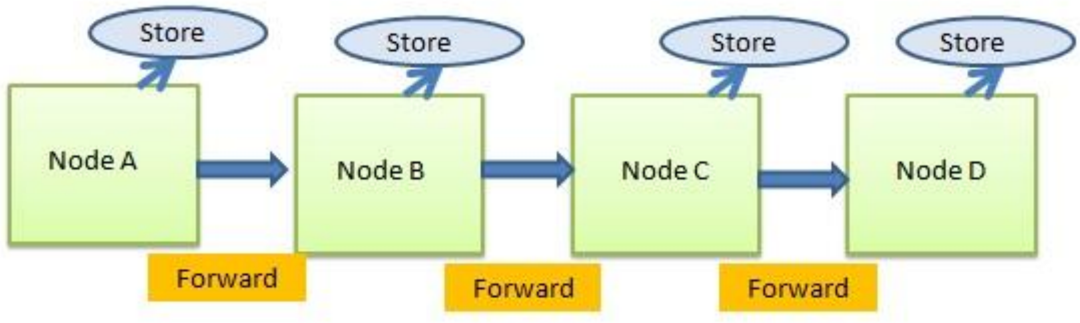


Fig.2. 1 Store and Forward Technique

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 and buffers. Internet routers use memory chips and buffers 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. DTN routers need persistent storage for their queues for one or more of the following reasons:

1. A communication link to the next hop may not be available for a long time.
2. One node in a communicating pair may send or receive data much faster or more reliably than the other node.
3. A message, once transmitted, may need to be retransmitted if an error occurs at an upstream (toward the destination) node, or if an upstream node declines acceptance of a forwarded message.
4. 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.

In the DTN, the fundamental concept is an Architecture based on Internet – Independent Middleware, where the protocols at all layers are used that bets suite the operation within each environment, with a new overlay network called Bundle Protocol (BP) inserted between application & the locally optimized communication stacks. Military applications in the DTN areas are substantial, allowing the retrieval of critical information in mobile battlefield scenarios using only intermittently connected network communications. For these kinds of applications, the DTN protocol should transmit data segments across multi – hop networks that consists of different regional networks based on environmental network parameters. In all the cases, the operation requirements are differently altered and their performance is negatively altered rendering them Heterogeneous nature. The network uses variety of communication nodes, such as wireless, satellites, vehicle- mounted and unmanned aerial vehicle, to continuously advance message traffic even when there’s an obstacle in the path that would stop traffic in the traditionally network. The delay tolerant networks makes the network to continue its function reliably in the environment where communications are most challenging and most critical and the message traffic continues to flow despite geographical or structural or malicious disruptions.

The DTN Architecture is designed to effectively operate as an overlay on top of regional networks or as an Inter Planetary internet. Moreover, the Delay Tolerant Network can overcome problems characterized by Long – Delays, Asymmetric Data Rates, Intermittent Connectivity, High Error Rates due to extreme environments, distances encountered in Space communication at Inter–Planetary scale competently when compared with the traditional Internet suite.

2.1.4 Custody Transfers

DTNs support node-to-node retransmission of lost or corrupt data at both the transport and the bundle protocols. However, because no single transport protocol (the primary means of reliable transfer) typically operates end-to-end across a DTN, end-to-end reliability can only be implemented at the bundle layer. The bundle protocol supports node-to-node retransmission by means of *custody transfers* [15]. Such transfers are arranged between the bundle-protocol agents’ of successive nodes, at the initial request of the source application. When the current bundle custodian sends a bundle to the next custodian (not necessarily the next node in the path), it requests a custody transfer and starts a time-to-acknowledge retransmission timer. If the next bundle-protocol agent accepts custody, it returns 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 with a particular node. A bundle custodian must store a bundle 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 enhance end-to-end reliability, but they do not guarantee it. Further enhancement can be achieved by using both the custody transfer and return receipt services.

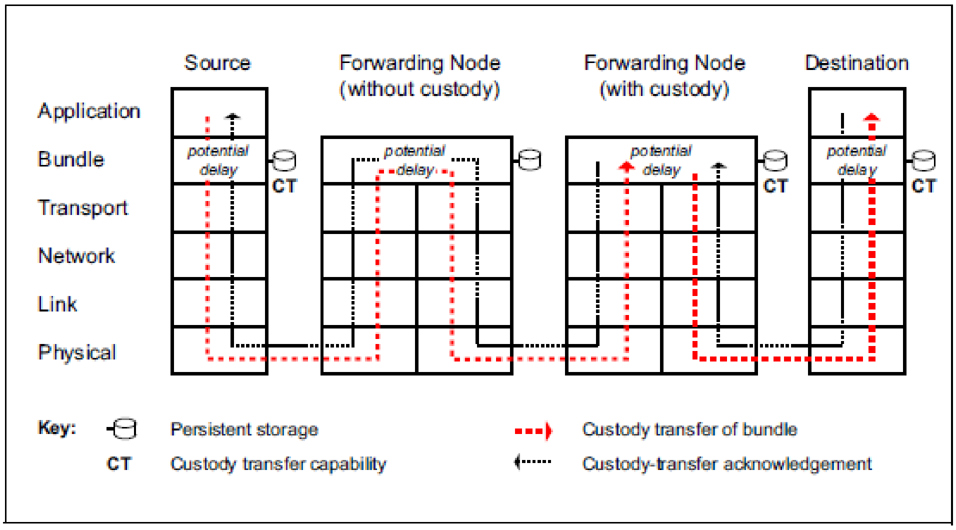


Fig.2. 2 Custody Transfer Mechanism of DTN

2.1.5 Applications of DTN

Although DTNs were originally conceived for interplanetary use, they may have a far greater number of applications on Earth. Here is a short summary of the possible applications [18]:

- **Space Agencies**: International Space Station communication (currently operational for research), interplanetary communication, future space-debris monitoring.

- **Military and Intelligence**: Mobile ad-hoc networks (MANETs) for wireless communication and monitoring, cargo tracking, search and rescue communication, unmanned aerial vehicle (UAV) communication and control.
- **Commercial**: Cargo and vehicle tracking (by road, rail, sea, and air), in-store and in-warehouse asset tracking, data transactions (e.g., financial, reservations), agricultural crop monitoring, processing-plant monitoring, communication in underground mines.
- **Public Service and Safety**: Security and disaster communication, search and rescue communication, humanitarian relief monitoring, smart-city event-response, smart transportation networks, smart electric-power networks, global airport-traffic control, infrastructure-integrity monitoring, unmanned aerial vehicle (UAV) communication and control, remote learning [9].
- **Personal Use**: Personal monitoring and communication in wilderness and urban areas, fire-and-forget text messaging. Environmental Monitoring: Animal migration, soil properties and stability, atmospheric and oceanographic conditions, seismological events.
- **Engineering and Scientific Research**: Network subject-matter experts, academic research by faculty and students.

2.2 Message Ferry

Message ferrying is a networking paradigm where a special node, called a message ferry, facilitates the connectivity in a mobile ad hoc network where the nodes are sparsely deployed. One of the key challenges under this paradigm is the design of ferry routes to achieve certain properties of end-to-end connectivity, such as, delay and message loss among the nodes in the ad hoc network. This is a difficult problem when the nodes in the network move arbitrarily. As we cannot be certain of the location of the nodes, we cannot design a route where the ferry can contact the nodes with certainty. Due to this difficulty, prior work has either considered ferry route design for ad hoc networks where the nodes are stationary, or where the nodes and the ferry move pro-actively in order to meet at certain locations. Such systems either require long-range radio or disrupt nodes’ mobility patterns which can be dictated by non-communication tasks. We present a message ferry route design algorithm that we call the Optimized Way-points, or OPWP, that generates a ferry route which assures good performance without requiring any online collaboration between the nodes and the ferry. The

OPWP ferry route comprises a set of way-points and waiting times at these way-points, that are chosen carefully based on the node mobility model. Each time that the ferry traverses this route, it contacts each mobile node with a certain minimum probability. The node-ferry contact probability in turn determines the frequency of node-ferry contacts and the properties of end-to-end delay. We show that OPWP consistently outperforms other naive ferry routing approaches.

2.2.1 Message Ferrying Model

In the Message Ferrying model, the devices in the network are classified into two categories [30].

(i) Regular nodes, or simply the nodes, that move according to some mobility model. These nodes generate data for other nodes in the network in the form of application layer data units called messages. At the same time, these nodes are interested in receiving the messages that other nodes have generated for them. For this work we assume that all the messages are unicast, i.e., they have a single unique destination. We assume that the movement of the nodes is driven by non-communication needs (e.g., a field-task assignment), and therefore this movement cannot be disrupted.

(ii) A single special node called message ferry (MF) that is responsible for delivering the messages between the nodes. The ferry achieves this by traversing a predetermined path repeatedly. We refer to each traversal through this route as a tour. We assume that both the ferry and the nodes are equipped with a similar radio of given small communication range. The nodes and ferry can communicate with each other only when they are within a distance of each other that is less than the communication range. The node and ferry are said to be in contact when they are within the communication range of each other.

We assume limited communication range because nodes may be energy constrained and may not be able to use long range communication channels that may require more power. Furthermore, while the ferry may be able to use a long range radio, the range of two-way communication between the node and the ferry would still be limited by the communication range of the nodes. Our model requires two-way communication for contact establishment. Similarly reliable data transfer between the nodes and the ferry, such as, using TCP, may also require two-way communication. During each successful contact, the ferry exchanges messages with the nodes. The ferry uploads the messages that the node has generated for other nodes, and downloads the messages that the ferry has for the particular node. The process is referred to as service. The ferry services only one node at a time. In the time between successive contacts, the nodes store the messages that they generate in a local buffer, called send buffer. We assume that the send buffer can hold a certain maximum number of messages, and once the send buffer is full, any new messages that the node generates are lost. We also assume that each node has a similar buffer for the messages that it receives; we call it the receive buffer. The receive buffer is used to store the messages that the node receives until they are consumed by the application layer at the node. We assume that the receive buffer for a node can hold a certain maximum number of messages. As the ferry takes the tour, it meets with different nodes. Upon meeting with a node, the ferry begins the download service, and continues until the ferry has downloaded all the messages that it has for the node, or, a timer, which we call the download timer, expires, or the receive buffer of the node becomes full, whichever occurs first. The ferry attempts to deliver any messages for the node that are left in its buffer at the end of download service in the next contact with the node. After the download service, the ferry starts the upload service. The ferry uploads the messages from the source buffer until all the messages in the source buffer of the node are uploaded, or a timer, called the upload timer, expires, whichever occurs first. Any messages that are left in the send buffer remain buffered until the next contact with the ferry. We refer to the messages that are left in the send buffer of the node as the residue messages. Please note that our model does not force strict order that the download service precede the upload service. These could happen simultaneously (if the radio channel permits), or in some multiplexed fashion. In general, performing download before upload reduces average delay, albeit slightly. Since both the ferry and the nodes are mobile in our model, we make a simplifying assumption that when the ferry and a node come in contact with each other, either the contact lasts long enough to complete the service, or they can pause, and exchange messages; usually the service time is short and does not amount to disruption in node mobility.

After the exchange the ferry continues with its route and the node continues with its movement. We assume that the ferry has infinite resources to move around and meet the other nodes, as well as to communicate with the other nodes when they come within its radio range, and to carry the messages between the nodes. Furthermore, we assume that we can route the message ferry in whatever way we want in the region where the nodes move. In this work the only constraints that we consider regarding the ferry are that the ferry cannot move faster than a certain maximum speed, and that it can only communicate with other nodes that are within a given radio range of the ferry.

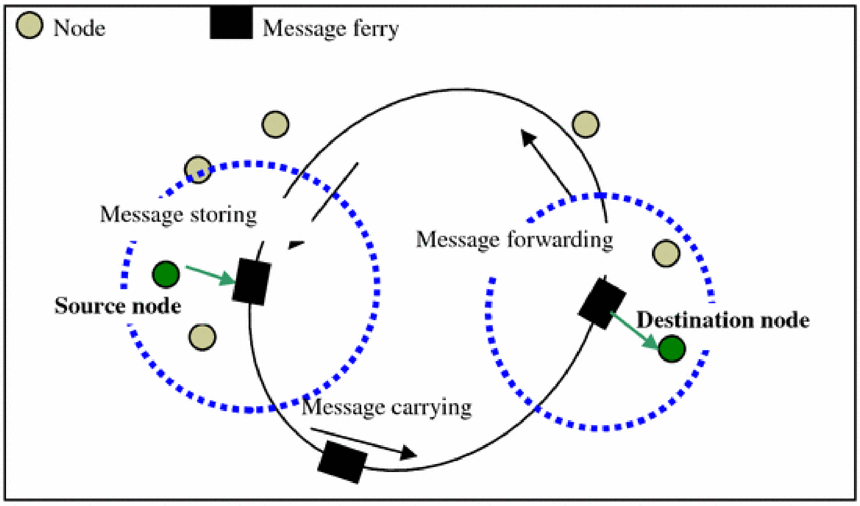


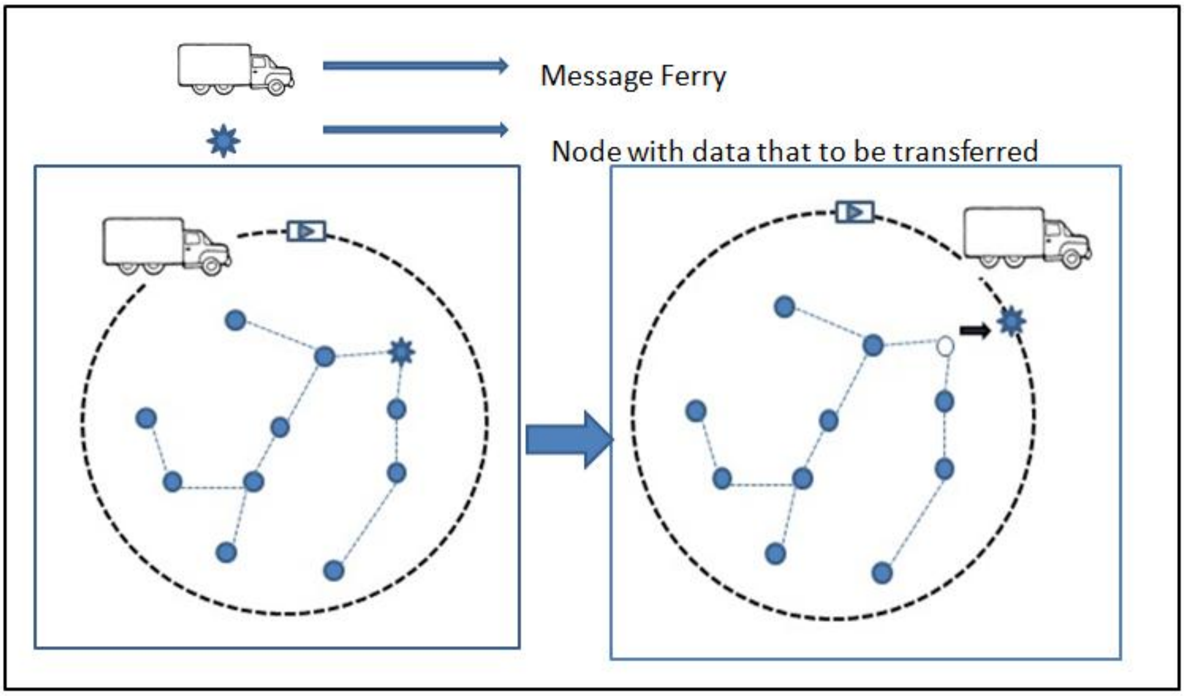
Fig.2. 3 Message Ferrying Model

2.2.2 Message Ferrying Scheme types

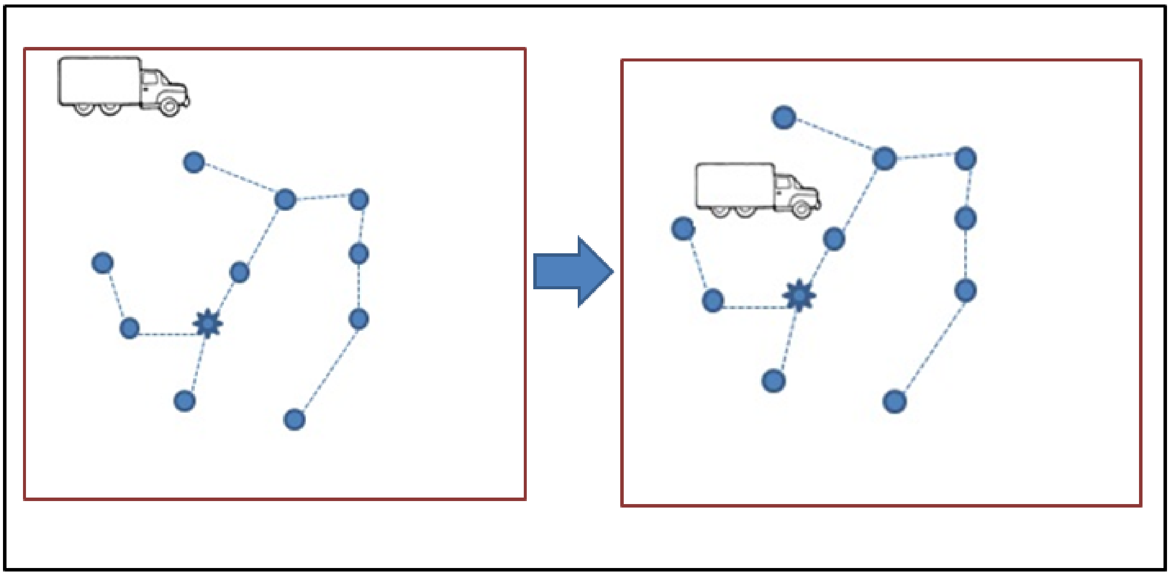
There are two message ferry schemes [1-4]: Node-initiated message ferry scheme and ferry-initiated message ferry scheme. In the node-initiated message ferry scheme, ferries move around the deployed area according to known routes and communicate with other nodes they meet. In this scheme, node requires 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 2.4(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 2.4(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 [1-4].

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, 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.



(a) Node Initiated Message Ferry: Message Ferry travels in a fixed known route and nodes move close to message ferry routes to transfer data (Here nodes are mobile)



(b) Ferry Initiated Message Ferry: Message ferry visits each node that requests to transfer data (Here nodes are generally static)

Fig.2. 4 Message Ferrying Scheme Types

2.3 Multi-Cluster DTNs

In 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 as *sink nodes* [1-4]. 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 kinds 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 2.4. We called this scenario as *ferry-assisted multi-cluster DTNs*. This network architecture is suitable for wide area sensing, e.g. Data MULE [48] 2. Note that in this kind of scenario, message ferries are equipped with 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, i.e., duration between message ferry 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 [1-4, 15] 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

2.4 Conclusion

After finishing this chapter we know that, DTN provides connectivity where internet can’t provide any end to end connectivity. This chapter briefly describes about Delay Tolerant Network (DTN) with its history, Store and Carry forward switching, Custody Transfer, Application of DTN, Message ferrying scheme and model and Multi Cluster DTN.

Next chapter we will discuss about Self Organized Data Aggregation Techniques and proposed model scenario.

Chapter 3

Self-organized Data Aggregation Technique

Last chapter we discussed about Delay Tolerant Network, History of Delay Tolerant Network and various use of Delay Tolerant Network. In the present chapter we will discuss about Self-Organized Data Aggregation Techniques, the intra-cluster communication, where, a fixed sink node collects bundles from nodes in isolated clusters with the help of the message ferry, as shown in Figure 3.1. 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.

3.1 Self-organized data aggregation technique: Network Model Scenario

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 [8] 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, [1-4] 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 .

To accomplish such a system, evolutionary game theoretic approach becomes one of the most appropriate mechanisms. The details of Evolutionary Game Theory will be discussed in Chapter 4

Evolutionary game theory originally explores the dynamics of a population of players under the influence of natural selection 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. 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 by taking account of the challenges.

With the help of the evolutionary game theoretic approach, the system [1-4] 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 chapter 5.

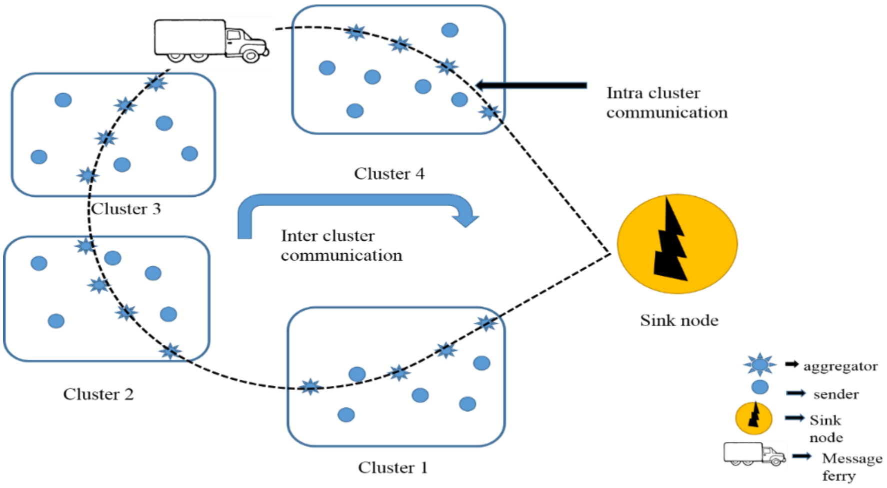


Fig.3. 1 Model Scenario: Message ferry visits a limited number of aggregators in each cluster and delivers collected bundles to sink node (Inter Cluster)

In system discussed [1-4], 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 visiting only the aggregators as illustrated in Figure 3.1.

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.

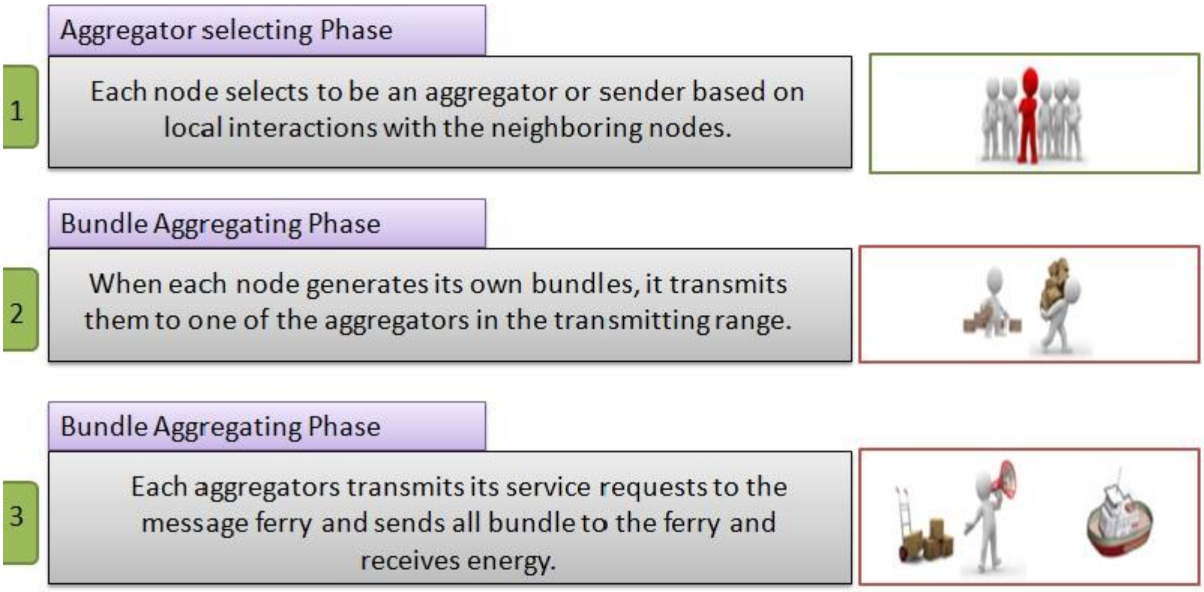
These service request messages contain the information of each aggregator's location and the amount of bundles it wants to transfer. To guide the message ferry, aggregators occasionally transmit location update messages. On reception of each information, the message ferry calculates the intra-cluster 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 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:

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.



Each node selects to be an aggregator or sender based on local interactions with the neighboring nodes.

When each node generates its own bundles, it transmits them to one of the aggregators in the transmitting range.

Each aggregators transmits its service requests to the message ferry and sends all bundle to the ferry and receives energy.

Fig.3. 2 Phases

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 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.

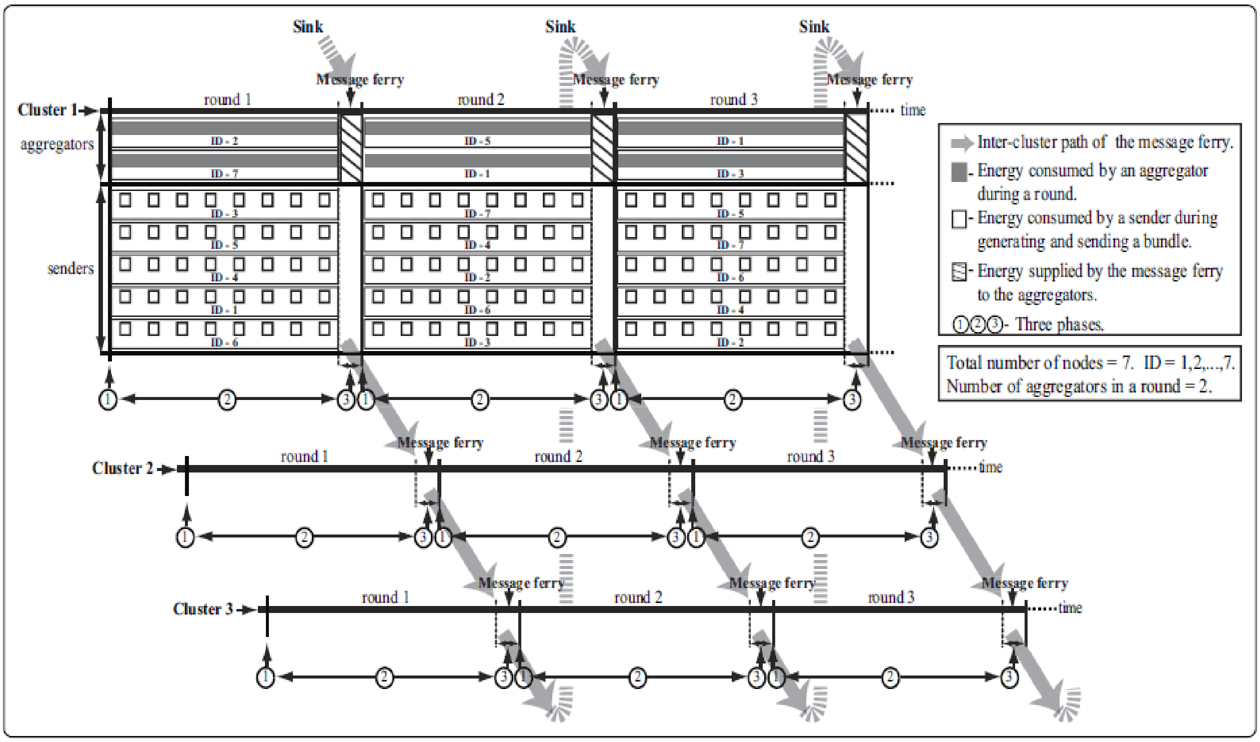


Fig.3. 3 Intra and Inter Cluster path of message ferry

As shown in Figure 3.3, 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 3.3 presents the above characteristics of aggregators and senders.

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

2) The role of a node should change per round. This can be shown in Figure 3.3. 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.

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.

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 3.3 and presented detailed in Figure 3.4, 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 3.4).

Let c and s denote the amount of energy consumption per round for aggregators 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 represents 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 3.4 illustrates the node level behavior for no-retransmission case and retransmission case. The details of node level behavior will be discussed in section 5.1

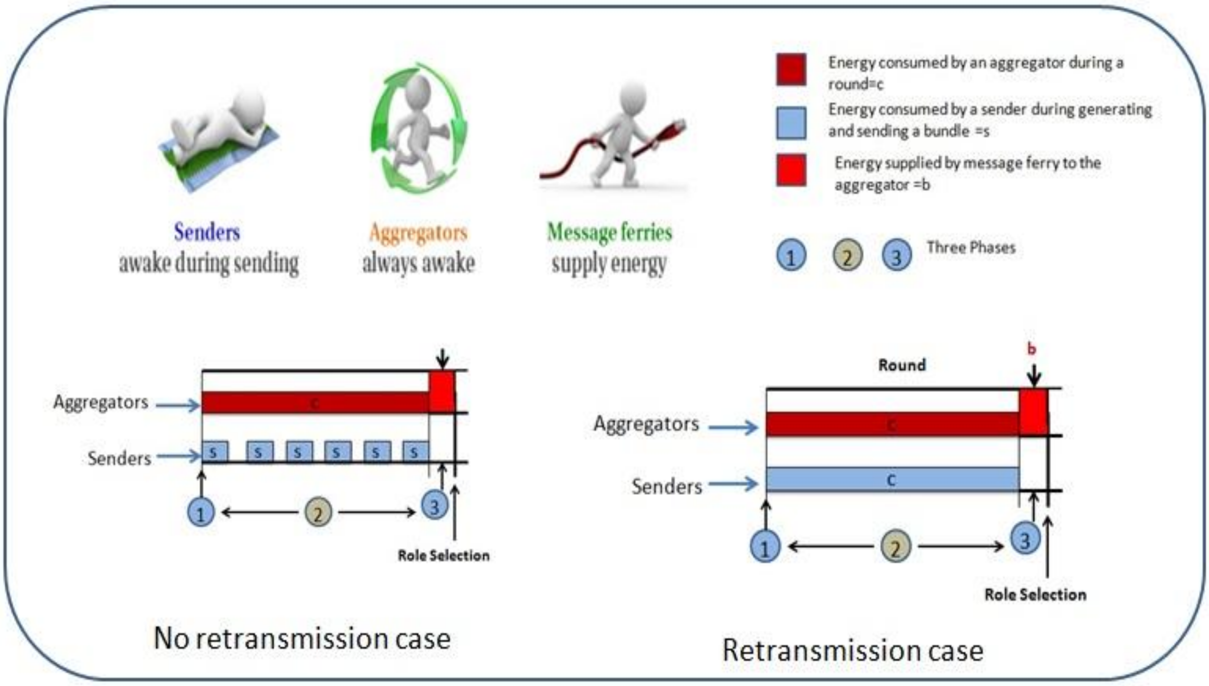


Fig. 3. 4 Role and node level energy consumption of an aggregator and a sender in one round. During the role selection period all nodes wakeup for a short time and select their next role based on their current conditions.

3.3 Conclusion

After above discussion, we know that, message ferry visits limited number of aggregators to collects bundles and deliver it to the sink nodes with Self Organized Data Aggregation Techniques. This chapter describes Self Organized Data Aggregation Techniques, Network Model scenario, three phases of network model, Intra and Inter cluster visiting path of message ferry and strategy selection of aggregators.

In next chapter, we discuss details about Evolutionary Game theory by which we can solve Self-Organized Data Aggregation techniques

Chapter 4

Game Theory, Evolutionary Game Theory and Replicator Equation on Graph

Last Chapter we discussed about Self-Organized Data Aggregation Techniques. This Self-Organized Data Aggregation Techniques can be solved and modeled as a game by Evolutionary Game Theory. We will use Replicator Equation on Graph of Evolutionary Game Theory for this purpose. So, in the present Chapter we will discuss about Game Theory, Evolutionary Game Theory, Replicator Equation and Replicator Equation on Graph in details.

4.1 Game Theory

Game theory [44, 45] is the study of strategic decision-making. It is "the study of mathematical models of conflict and cooperation between intelligent rational decision-makers. Game theory is mainly used in economics, political science and psychology, as well as logic, computer science, and biology. Originally, it addressed zero-sum games, in which one person's gains result in losses for the other participants. Today, game theory applies to a wide range of behavioral relations, and is now an umbrella term for the science of logical decision making in humans, animals, and computers.

Modern game theory began with the idea regarding the existence of mixed-strategy equilibria in two-person zero-sum games and its proof by John von Neumann. Von Neumann's original proof used Brouwer fixed-point theorem on continuous mappings into compact convex sets, which became a standard method in game theory and mathematical economics. His paper was followed by the 1944 book *Theory of Games and Economic Behavior,* co-written with Oskar Morgenstern, which considered cooperative games of several players. The second edition of this book provided an axiomatic theory of expected utility, which allowed mathematical statisticians and economists to treat decision-making under uncertainty.

This theory was developed extensively in the 1950s by many scholars. Game theory was later explicitly applied to biology in the 1970s, although similar developments go back at least as far as the 1930s. Game theory has been widely recognized as an important tool in many fields. With the Nobel Memorial Prize in Economic Sciences going to game theorist Jean Tirole in 2014, eleven game-theorists have now won the economics Nobel Prize. John

Maynard Smith was awarded the Crafoord Prize for his application of game theory to biology.

4.1.1 Nash Equilibrium

John Nash: One of the early researchers in game theory developed an equilibrium further named after him.

In Game Theory, the Nash equilibrium is a solution concept of a non-co-operative game involving two or more players, in which each player is assumed to know the equilibrium strategies of the other players, and no player has anything to gain by changing only their own strategy. If each player has chosen a strategy and no player can benefit by changing strategies while the other players keep theirs unchanged, then the current set of strategy choices and the corresponding payoffs constitutes a Nash equilibrium. The reality of the Nash equilibrium of a game can be tested using experimental economics method.

Stated simply, Amy and Will are in Nash equilibrium if Amy is making the best decision she can, taking into account Will's decision while Will's decision remains unchanged, and Will is making the best decision he can, taking into account Amy's decision while Amy's decision remains unchanged. Likewise, a group of players are in Nash equilibrium if each one is making the best decision possible, taking into account the decisions of the others in the game as long the other party's decision remains unchanged.

4.1.2 Applications

Game theorists use the Nash equilibrium concept to analyze the outcome of the strategic interaction of several decision makers. In other words, it provides a way of predicting what will happen if several people or several institutions are making decisions at the same time, and if the outcome depends on the decisions of the others. The simple insight underlying John Nash's idea is that one cannot predict the result of the choices of multiple decision makers if one analyzes those decisions in isolation. Instead, one must ask what each player would do, taking into account the decision-making of the others.

Nash equilibrium has been used to analyze hostile situations like war and arms races and also how conflict may be mitigated by repeated interaction .It has also been used to study to what extent people with different preferences can cooperate, and whether they will take risks to achieve a cooperative outcome It has been used to study the adoption of technical standards, and also the occurrence of bank runs and currency crises. Other applications include traffic flow , how to organize auctions ,outcome of efforts exerted by multiple parties in the education process, regulatory legislation such as environmental regulations, and even penalty kicks in football .

4.2 Evolutionary Game Theory

Evolutionary game theory (EGT) is the application of game theory to evolving populations of life forms in biology. EGT is useful in this context by defining a framework of contests, strategies, and analytics into which Darwinian competition can be modeled. EGT originated in 1973 with John Maynard Smith and George R. Price's formalization of the way in which such contests can be analyzed as "strategies" and the mathematical criteria that can be used to predict the resulting prevalence of such competing strategies.

Evolutionary game theory differs from classical game theory by focusing more on the dynamics of strategy change as influenced not solely by the quality of the various competing strategies, but by the effect of the frequency with which those various competing strategies are found in the population.

Evolutionary game theory has proven itself to be invaluable in helping to explain many complex and challenging aspects of biology. It has been particularly helpful in establishing the basis of altruistic behaviors within the context of Darwinian process. Despite its origin and original purpose, evolutionary game theory has become of increasing interest to economists, sociologists, anthropologists, and philosophers.

The birth of evolutionary game theory is marked by the publication of a series of papers by mathematical biologist John Maynard Smith [48]. Maynard Smith adapted the methods of traditional game theory [44, 45], which were created to model the behavior of rational economic agents, to the context of biological natural selection. He proposed his notion of an evolutionarily stable strategy (ESS) as a way of explaining the existence of ritualized animal conflict. Maynard Smith’s equilibrium concept was provided with an explicit dynamic foundation through a deferential equation model introduced by Taylor and Jonker. Schuster and Sigmund [48,49], following Dawkins [47-49], dubbed this model the replicator dynamic, and recognized the close links between this game-theoretic dynamic and dynamics studied much earlier in population ecology and population genetics. By the 1980s, evolutionary game theory was a well-developed and firmly established modeling framework in biology.

Towards the end of this period, economists realized the value of the evolutionary approach to game theory in social science contexts, both as a method of providing foundations for the equilibrium concepts of traditional game theory, and as a tool for selecting among equilibrium in games that admit more than one. Especially in its early stages, work by economists in evolutionary game theory hewed closely to the interpretation set out by biologists, with the notion of ESS and the replicator dynamic understood as modeling natural selection in populations of agents genetically programmed to behave in specific ways. But it soon became clear that models of essentially the same form could be used to study the behavior of populations of active decision makers Indeed, the two approaches sometimes lead to identical models: the replicator dynamic itself can be understood not only as a model of natural selection, but also as one of imitation of successful opponents .

While the majority of work in evolutionary game theory has been undertaken by biologists and economists, closely related models have been applied to questions in a variety of fields, including transportation science computer science, and sociology . Some paradigms from evolutionary game theory are close relatives of certain models from physics, and so have attracted the attention of workers in this field. All told, evolutionary game theory provides a common ground for workers from a wide range of disciplines.

Evolutionary game theory (EGT) [44, 45] has grown into a field that combines the principles of game theory, evolution, and dynamical systems to interpret the interactions of biological agents. Practitioners in the field have used the theory to explain biological phenomena successfully, but EGT can also be used to interpret classical games from a different perspective. This document introduces evolutionary game theory and presents an evolutionary approach to the analysis of games. There are several basic components in the EGT analysis of games. Game agents and their strategies must be simulated with populations of players, the fitness of different strategies relative to the population must be computed, and a process to govern the evolution of the population must be defined. These simple components can be combined to yield highly complex solutions. Ideally, under the dynamical process the strategies of the populations of players will converge to some stable value.

Evolutionary game theorists often claim the evolutionary solution of the game as the true definition of rational play.

Evolutionary game theory studies the behavior of large populations of agents who repeatedly engage in strategic interactions. Changes in behavior in these populations are driven either by natural selection via differences in birth and death rates, or by the application of myopic decision rules by individual agents.

When there are evolutionary conflicts between individuals (such as competition for resources) the best strategy to use may depend upon strategies used by other competitors. Game theory is used to make theoretical models of these situations which are usually subject to frequency dependent selection where the fitness of a morph varies with its frequency in the population.

Game theory is applicable only for two players. When this play will be played with infinite number of players, the game is going to evolve to a particular player .Then the game theory will be called Evolutionary Game Theory.

4.3 Replicator Equation

The natural selection process that determines how populations playing specific strategies evolve is known as the replicator dynamics. Slightly differing versions of these equations can be found in .There are different replicator dynamics depending on the evolutionary model being used.

The replicator equation [45-47] is the first and most important game dynamics studied in connection with evolutionary game theory. It was originally developed for symmetric games with finitely many strategies.

Consider an evolutionary game with n strategies, labeled $i=1,.....n$. The payoff matrix, A, is an $n \times n$ matrix, whose entries, $A = [a_{ij}]$, denote the payoff for strategy i versus strategy j. The relative abundance (frequency) of each strategy is given by x_i. We have

$$\sum_{i=1}^{n} x_i = 1 \tag{4.1}$$

The fitness of strategy *i* is given by

$$f_i = \sum_{j=1}^{n} x_j a_{ij} \tag{4.2}$$

For the average fitness of the population, we obtain

$$\Phi = \sum_{i=1}^{n} x_i f_i \tag{4.3}$$

The replicator equation is given by

$$\dot{x_i} = x_i(f_i - \Phi), i = 1; \ldots n. \tag{4.4}$$

This equation is one of the fundamental equations of evolutionary dynamics where dot represent the time derivatives. It describes evolutionary game dynamics (frequency dependent selection) in the deterministic limit of an infinitely large, well-mixed population.

Note that Equation (4.1) is applicable only to an infinitely large and well-mixed population where each player can equally play games with all other nodes.

4.4 Replicator Equation on graph

Replicator Equation on graphs 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.

Let us introduce the $n \times n$ matrix also known as modifier matrix M= $[m_{ij}]$ (where m_{ij} describes the local competition between strategy *i* and *j*)

Note that Off-diagonal elements of matrix m is anti-symmetric, i.e., $m_{ij} = -m_{ji}$, that is ,

Table 4. 1: Modifier Matrix

Node 2 / Node 1	Send	Aggregate
Send	**0,0**	***m,-m***
Aggregate	***-m, m***	**0,0**

because the gain of one strategy in local competition is the loss of another. Further, diagonal elements m_{ii} and m_{jj} are always zero, suggesting that local competition between the same strategies results in zero.

By modifying the original pay-off matrix, a_{ij}, the Evolutionary game dynamics in a well-mixed population can be transformed into on a *k* regular graph. The modified pay off matrix,

$$A' = [a_{ij} + m_{ij}] \tag{4.5}$$

The expected payoff g_i for the local competition of strategy *i* is defined as

$$g_i = \sum_{j=1}^{n} x_j m_{ij} \tag{4.6}$$

Note that, the average pay-off of local competition of strategy of *i* sums to zero i.e,

$$\sum_{i=1}^{n} x_i g_i = 0 \tag{4.7}$$

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

$$\Phi = \sum_{i=1}^{n} x_i(f_i + g_i) = \sum_{i=1}^{n} x_i f_i \tag{4.8}$$

which is same as previous equation.

Let,x_i denote the frequency of strategy*i*on *k* regular graph. The replicator equation on graph can be obtained as follows:

$$\dot{x}_i = x_i(f_i + g_i - \Phi), i = 1; \dots n \tag{4.9}$$

where f_i, g_i, Φ is given in equation respectively.

Note that, as k increases, the relative contribution of g_i decreases compared to f_i and in the limit *k*→∞, the replicator equation on graph is reduce to regular replicator equation.

4.5 Conclusion

Previous chapter we described about Self Organized Data Aggregation Techniques which can be solved by the help of Evolutionary Game Theory which is described in present chapter.

Replicator Equation is used for infinite number of players and Replicator Equation on graph is for finite number of players .That’s why, in our thesis, we use Replicator Equation on Graph. In summary, this chapter briefly describes about Game Theory, Nash Equilibrium, Applications of game theory and Nash Equilibrium, Evolutionary Game Theory, Replicator Equation on Graph and Modifier Matrix.

Next Chapter we will discuss our proposed mathematical model of three update rules that is Birth-Death Update, Death-Birth Update and Imitation Update rules.

Chapter 5

Mathematical Model of Proposed Scenario

Last chapter we discussed about Game theory, Evolutionary Game Theory, Replicator Equation and Replicator Equation on Graph. Note that, Game theory is applicable only for two players where Evolutionary Game theory is applicable for Infinite population and Replicator Equation on Graph is used for finite number of population. Now, in this chapter, we will discuss about mathematical model of three Update Rules of Evolutionary Game Theory.

5.1 Pay-off Matrix

Now, the interaction among nodes can be modeled as a game between two neighboring nodes in evolutionary game theory [5], which is represented by a payoff matrix. In our scenario, there are two strategies for each node: to become an aggregator (aggregate) and a sender (send). There are four possible combinations of the strategies of the two nodes, and payoff of each node depends on the combination of strategies.

Table 5. 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 5. 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 5. 3: Abstract payoff matrix

Node 2 / Node 1	Sender	Aggregator
Sender	***R, R***	***S, T***
Aggregator	***T, S***	***P, P***

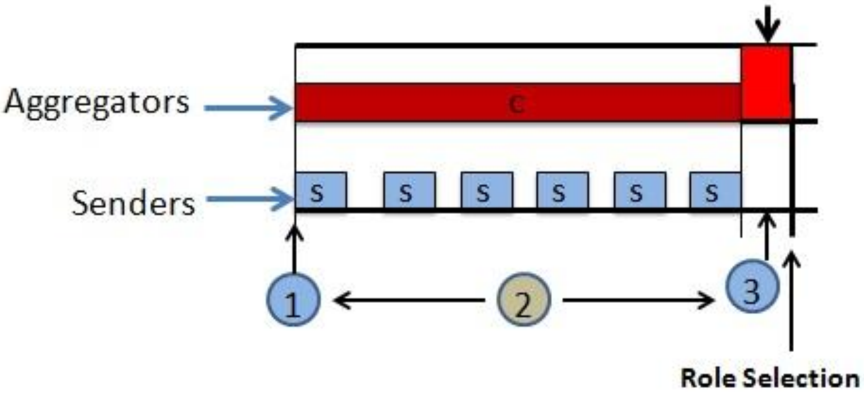
Let c and s denote the amount of energy consumption for aggregators and senders, respectively, per round. Obviously, $c \geq s$. s approaches c with the rate of generating bundles. The energy supplied by the message ferry to each aggregator is represented by b. intuitively, the larger b is, the more the aggregators increases. $b \geq c$ should also be satisfied to suppress the number of senders.

In summary, the benefit b is only a parameter for the sink node to control the number of aggregators in each cluster and should be carefully tuned by the sink node. We first model the bargain among nodes as a game between two neighboring nodes in evolutionary game theory. There are two roles (strategies) for each node: aggregator (aggregate) and sender (send). There are four possible combinations of the strategies of the two nodes as in Table 5.1.

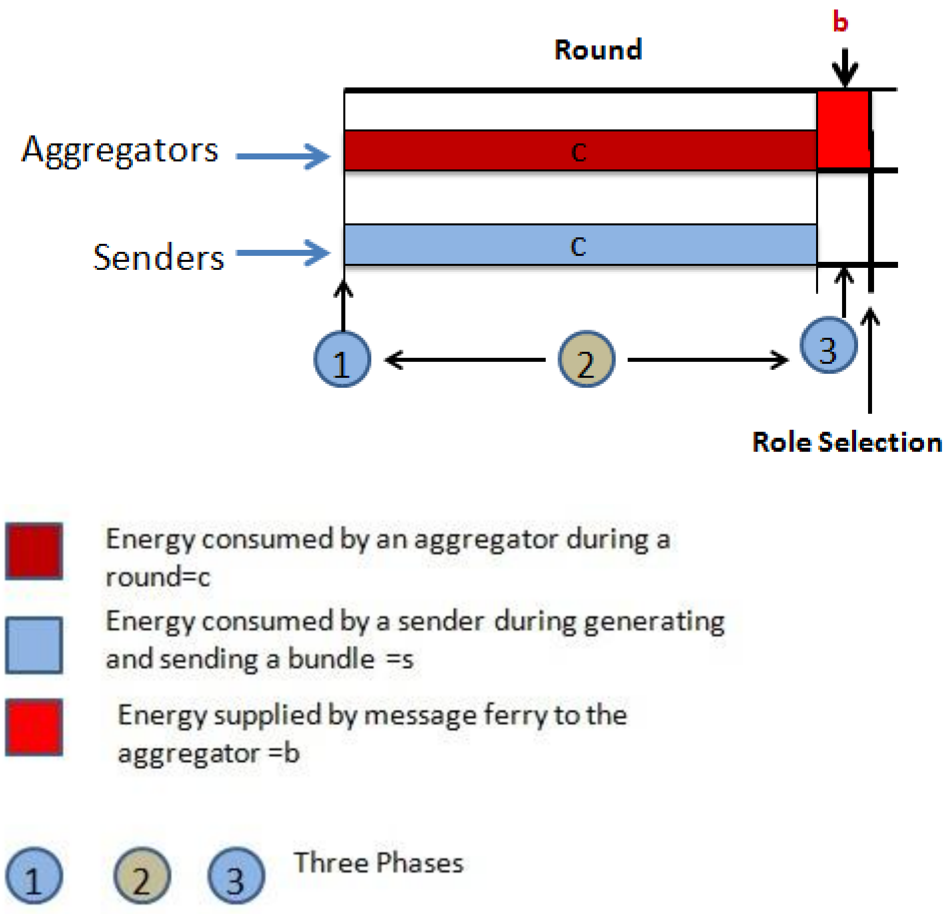
The resulting payoffs for each case can be modeled by taking the energy supply and energy consumption into account. If both nodes select to be aggregators, they lose the largest energy c without any energy supply from the message ferry, because they are not be able to collect a sufficient number of bundles to request the ferry to visit. An aggregator paired with a sender obtains the largest energy $b-c$; it loses c but obtains b from the message ferry. In this case, the corresponding sender loses the smallest energy s. In the case, when both nodes select to be senders and consume s (no retransmission). Here, they lose c in the worst case where the sender has to keep awake all the time in a round due to continuous retransmission.

We can abstract Table 5.1 into Table 5.2 and Table 5.3 where $T > S = R > P$ (no retransmission) or $T > S > R = P$ (retransmission). In both cases, every node not only has a temptation to be an aggregator ($T > R$) but also a fear to be an aggregator ($S > P$). The larger b is, the more the temptation is. This indicates that the sink node can control the number of aggregators (senders) by changing b.

The condition $T > R$ and $S > P$ also has another significant characteristic; taking a strategy different from the opponent is better than taking the same strategy as the opponent. As a result, both aggregating and sending strategies stably coexist [5]. Thus, with the help of the payoff-matrix and evolutionary game theory, when each node undertakes suitable strategies to optimize its own payoff, then the system converges to a fully stable situation where both senders and aggregators stably coexist.



a) Energy consumption for no retransmission case



b) Energy consumption for retransmission case

Fig.5. 1 Energy consumption

First we derive the replicator equation on graph for BD, DB and IM update rule by using payoff matrix from Table 5.1 and Table 5.2 for no retransmission case and retransmission case respectively.

5.2 Three different update rules of Evolutionary Game Theory

We consider three different update rules for the evolutionary dynamics, which we call ‘Birth–Death’ (BD), ‘Death–Birth’ (DB) and ‘imitation’ (IM).

I. **Birth-Death update rule (BD):** A node (called mother node) is selected depending on its own payoff to update the strategy of randomly chosen neighboring node (called offspring node). The new updated strategy of the offspring node is changed similar to the mother node (which implies mother node is giving birth of its strategy).While updating, the previous strategy of offspring is deleted (which implies the strategy of offspring node is death). In Fig.5.2 node 1 is selected (here it is called mother node) to update the strategy of randomly chosen neighboring node 2 (here it is called off spring node) in (a).The new updated strategy of the offspring node is changed similar to the mother node (which implies mother node is giving birth of its strategy) in (b).While updating, the previous strategy of offspring is deleted (which implies the strategy of offspring node is death).

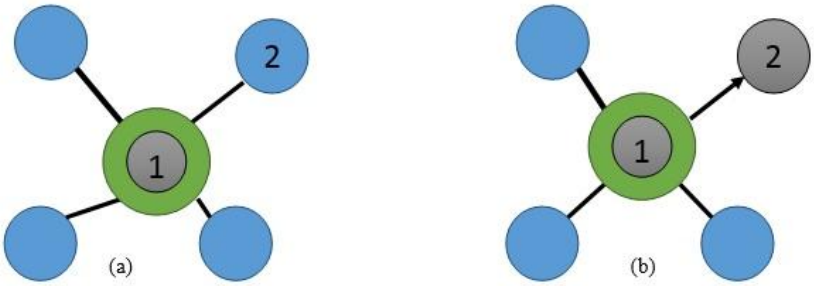


Fig.5. 2 Strategy Birth-Death update rule (BD)

II. **<u>Death-Birth Update Rule (DB)</u>:** A node (here it is offspring node) is selected depending on its pay off to delete its own existing strategy (which implies offspring node is giving death of its existing strategy).The strategy of offspring node is renewed and made similar to the strategy of a selected node (mother node which gives birth of it's strategy) which is chosen among neighboring nodes depending on its payoff. In Fig.5.3 node 1 (here it is off spring node) is selected to delete its current strategy (which implies offspring node is giving death of its existing strategy) in (a). The strategy of offspring node is renewed and made similar to the strategy of a selected node (mother node which gives birth of it's strategy) which is chosen among neighboring nodes depending on its payoff which is shown in (b)

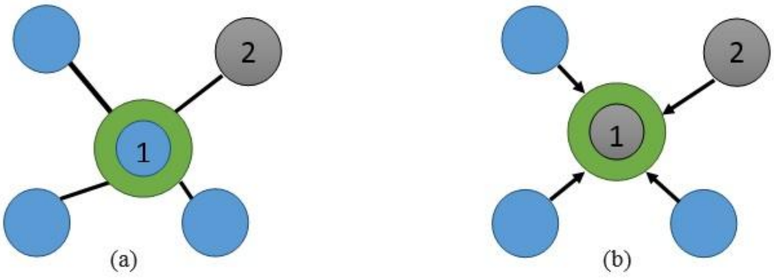


Fig.5. 3 Strategy Death-Birth update rule (DB)

III. **<u>Imitation Update Rule (IM)</u>:** Each node analyses 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 present strategy or imitating one of the neighbor's strategies proportional to payoffs. In Fig.5.4 node 1 analyses its own payoffs according to the strategies of its neighbors (here it is node 2), as well as its own strategy, in the previous round (shown in figure a). It then chooses between keeping its present strategy (which is shown in figure b) or imitating one of the neighbor's strategies proportional to payoffs (which is shown in figure c).

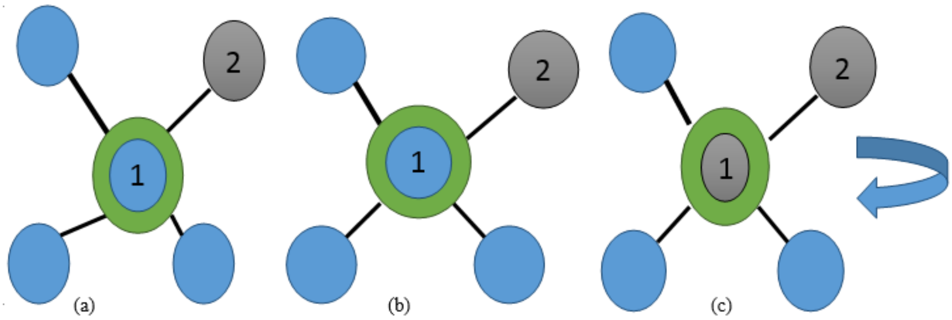


Fig.5. 4 Strategy Imitation update rule (IM)

5.3 Mathematical Model of our proposed scenario

Let x denote the ratio of the number of aggregators to the total number of cluster members. Note that $1 - x$ represents the ratio of the number of senders. For the no retransmission case, we use Table 5.1.

$x_1 = x$, denotes the ratio of Aggregators and $x_2 = 1 - x$, denotes the ratio of senders.

5.3.1 <u>Birth Death- Update Rule (BD) for the case of no-retransmission</u>

The Modifier Matrix for the case of BD is given by [46-48]

$$m_{ij(BD)} = \frac{a_{ii+}a_{ij-}a_{ji-a_{jj}}}{k-2} \tag{5.1}$$

The expected payoff (fitness) $f_{1(BD)}$ and $f_{2(BD)}$ of aggregators and senders can be obtained by substituting the value of Table 5.1 into Eq. (4.2)

$$f_{1(BD)} = x(-c) + (1-x)(b-c), \qquad (5.2)$$
$$f_{2(BD)} = -s$$

respectively.

The expected pay-off for local competition is obtained by putting the values of Table 5.1 (for the case of no retransmission) into Eq. (4.6)

$$g_{1(BD)} = (1-x)m, \qquad (5.3)$$
$$g_{2(BD)} = -xm$$

respectively.

The modifier matrix can be obtained by substituting the value of Table 5.1 into Eq. (5.1)

$$m_{ij(BD)} = \frac{b-2c+2s}{k-2} \qquad (5.4)$$

The average pay off can be obtained by using Eq. (4.8) is

$$\Phi_{(BD)} = x_1\ [f_{1(BD)}\ + g_{1(BD)}\ ) + x_2\ (f_{2(BD)}\ + g_{2(BD)}\ ] \qquad (5.5)$$
$$= (\ bx - s)(1-x) - cx$$

Finally the replicator equation on graph to become an aggregator is obtained by using Eq. (4.9)

$$\dot{x_1} = x_1[f_{1(BD)} + g_{1(BD)} - \Phi_{(BD)}] \qquad (5.6)$$
$$= x(1-x)[\frac{-k(c-s) + bk(1-x) + b(2x-1)}{k-2}]$$

Substituting $\dot{x_1}$=0, there exists three equilibria $x = 0{,}1$ and

$$x_{1(BD)}^{*} = \frac{b(k-1) - k(c-s)}{b(k-2)} \qquad (5.7)$$

Note, equilibria is feasible if 0<x<1, i.e,

$$\frac{k}{k-1} < \frac{b}{c-s} < k \qquad (5.8)$$

holds, where we use $c - s$ >0. We also have for all k >2, $0 < \frac{k}{k-1} < k$. As a result, for any c, s, and k, there exists b >0. Thus the equilibria is controllable and it can be shown to be stable.

5.3.2 Birth Death- Update Rule (BD) for the case of retransmission

Similarly, for the retransmission case (The detailed calculation is discussed in Appendix A) with the help of pay-off matrix stated in Table 5.2, the stable and controllable equilibrium becomes

$$x^*_{2(BD,Re)} = \frac{b(k-2)+b-(c-s)}{b(k-2)+(k-2)(c-s)} \tag{5.9}$$

Note, equilibria is feasible if 0<x<1, i.e,

$$\frac{1}{k-1} < \frac{b}{c-s} < k-1 \tag{5.10}$$

holds, where we use $c - s$ >0. We also have for all k >2, $0 < \frac{1}{k-1} < k-1$. As a result, for any c, s, and k, there exists b >0. Thus the equilibria is controllable and it can be shown to be stable.

5.3.3 Death-Birth Update Rule (DB) for the case of no-retransmission

The stable and controllable equilibrium for DB update rule for the case of no-retransmission is (The detailed calculation is discussed in Appendix B)

$$x^*_{1(DB)} = \frac{b(k^2-k-2)-k^2(c-s)+b}{b(k^2-k-2)} \tag{5.11}$$

The equilibria is feasible if 0<x<1, i.e,

$$\frac{k^2}{k^2-k-1} < \frac{b}{c-s} < k^2 \tag{5.12}$$

where, we use $c - s$ >0. We also have for all k >2, $0 < \frac{k^2}{k^2-k-1} < k^2$. As a result, for any c, s, and k, there exists b >0. Thus the equilibria is controllable and it can be shown to be stable.

5.3.4 Death-Birth Update Rule (DB) for the case of retransmission

The stable and controllable equilibrium for DB update rule for the case of Retransmission is (The detailed calculation is discussed in Appendix C)

$$x^*_{2(DB,Re)} = \frac{b(k+1)(k-2)+b-(c-s)}{(k-2)(k+1)(b+c-s)} \tag{5.13}$$

The equilibria is feasible if 0<x<1, i.e.

$$\frac{1}{k^2-k-1} < \frac{b}{c-s} < k^2-k-1 \tag{5.14}$$

holds, where we use $c - s > 0$. We also have for all $k > 2$, $0 < \frac{1}{k^2-k-1} < k^2-k-1$. As a result, for any *c, s,* and *k*, there exists $b > 0$. Thus the equilibria is controllable and it can be shown to be stable.

5.3.5 Imitation Update Rule (IM) for no-retransmission case

The stable and controllable equilibrium for IM update rule for the case of no-retransmission is (The detailed calculation is discussed in Appendix D)

$$x^*_{1(IM)} = \frac{b(k^2+k-3)-(k^2+2k)(c-s)}{b(k+3)(k-2)} \tag{5.15}$$

The equilibria is feasible if 0<x<1, i.e.

$$\frac{k^2+2k}{k^2+k-3} < \frac{b}{c-s} < \frac{k^2+2k}{3} \tag{5.16}$$

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

5.3.6 Imitation Update Rule (IM) for the case of retransmission

The stable and controllable equilibrium for IM update rule for the case of no-retransmission is (The detailed calculation is discussed in Appendix E)

$$x^*_{2(IM,Re)} = \frac{b(k^2+k-3)-3(c-s)}{(k-2)(k+3)(b+c-s)} \tag{5.17}$$

Note, equilibria is feasible if 0<x<1, i.e,

$$\frac{3}{k^2+k-3} < \frac{b}{c-s} < \frac{k^2+k-3}{3} \tag{5.18}$$

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

5.4 Summary for three update rules for two cases

Now we summarize the above mathematical model of our proposed scenario in following tables:

5.4.1 For the case of no-retransmission

Table 5.4: Solution of Replicator Equation on Graph for no-retransmission

Update rules	Ratio of Aggregators	Eq. No
Birth-Death	$x^*_{1(BD)} = \frac{b(k-1) - k(c-s)}{b(k-2)}$	(5.7)
Death-Birth	$x^*_{1(DB)} = \frac{b(k^2 - k - 2) - k^2(c-s) + b}{b(k^2 - k - 2)}$	(5.11)
Imitation	$x^*_{1(IM)} = \frac{b(k^2 + k - 3) - (c-s)(k^2 + 2k)}{b(k+3)(k-2)}$	(5.15)

Table 5.5: Equilibrium Condition for three cases of no-retransmission

Update rules	Equilibrium Condition	Eq. No
Birth-Death	$\frac{k}{k-1} < \frac{b}{c-s} < k$	(5.8)
Death-Birth	$\frac{k^2}{k^2-k-1} < \frac{b}{c-s} < k^2$	(5.12)
Imitation	$\frac{k^2+2k}{k^2+k-3} < \frac{b}{c-s} < \frac{k^2+2k}{3}$	(5.16)

5.4.2 For the case of retransmission

Table 5. 6: Solution of Replicator Equation on Graph for retransmission

Update rules	Ratio of Aggregators	Eq. No
Birth-Death	$x^*_{2(BD,Re)} = \dfrac{b(k-2)+b-(c-s)}{b(k-2)+(k-2)(c-s)}$	(5.9)
Death-Birth	$x^*_{2(DB,Re)} = \dfrac{b(k+1)(k-2)+b-(c-s)}{(k-2)(k+1)(b+c-s)}$	(5.13)
Imitation	$x^*_{2(IM,Re)} = \dfrac{b(k^2+k-3)-3(c-s)}{(k-2)(k+3)(b+c-s)}$	(5.17)

Table 5. 7: Equilibrium Condition for three cases of retransmission

Update rules	Equilibrium Condition	Eq. No
Birth-Death Update Rule	$\dfrac{1}{k-1} < \dfrac{b}{c-s} < k-1$	(5.10)
Death-Birth Update Rule	$\dfrac{1}{k^2-k-1} < \dfrac{b}{c-s} < k^2-k-1$	(5.14)
Imitation Update Rule	$\dfrac{3}{k^2+k-3} < \dfrac{b}{c-s} < \dfrac{k^2+k-3}{3}$	(5.18)

5.5 Criteria for selecting the best update rule for a particular case

From the above mathematical calculations and from the table above, we can see that, we have four independent variable *intensive (b), energy consumed by aggregators (c), energy consumed by senders (s)* and *degree (k).* These four variables will affect the *ratio of aggregators (x*).*Now we will plot these variables against *degree (k).*These following criterion should be followed when to select the best strategy for a particular case:

- It is better to choose small number of *Degrees (k).* Highest values of degree mean all nodes are connected to each other which are not possible in practical.

- When we plot *Degree (k)* vs. *Intensive (b)* curve, the update rule which will give the least intensive for the same values of degree, we will select that update rule is as better because our objective is to get less intensive supplied by message ferry to nodes to become aggregators.
- When we plot *Degree (k)* vs. *Modifier matrix (m)* curve for variable *intensive (b),* we will select that update rule as better which will give highest values of modifier matrix as highest values of modifier matrix means nodes are more willingly to become aggregators and senders.
- When we plot *Degree (k)* vs. *Ratio of aggregators (x*)* curve for variable *intensive (b),* the update rule is selected as better which will give least number of ratio of aggregators at less number of intensive supplied by message ferry.
- Again, when we plot *Degree (k)* vs. *Intensive (b)* curve for variable *Ratio of Aggregators (x*),*that update rule will be selected as better update rule which will give least intensive supplied by message ferry for less number of ratio of aggregators.

5.6 Conclusion

This chapter we discussed our proposed mathematical model. For this we discussed abstract payoff matrix, pay off matrix for the case no retransmission and retransmission case respectively, introduced valid parameters, energy consumption for two cases respectively, three update rule of evolutionary game theory and mathematical calculations for no retransmission and retransmission respectively and summary for three update rules.

Next Chapter we are going analysis and discuss the Numerical Analysis for three update rules.

Chapter 6

Numerical Analysis

Last Chapter we discussed the mathematical model of three Update rule for the case of no-retransmission and retransmission. In, this chapter we will discuss about the comparison of three update rules, observe the graphs and discuss about the facts.

6.1 Comparison for no-retransmission

We have four independent variables, intensive (*b)*, energy consumed by aggregators *(c)*, energy consumed by senders *(s)* and degree (*k)*.These for variables affect the ratio of aggregators (*x*)*. First, for simplification, we assume that *c-s*= 1. Note that this simplification does not lose generality.

As a result, the ratio of aggregators can be controlled only by *b* and *k* according to Eq. (5.8), (5.12) and (5.16) for the case no-retransmission and Eq. (5.10), (5.14) and (5.18) respectively. The expected number of aggregators can be obtained by the product of *x* and the number of cluster members.

.

6.1.1 *Degree (k)* vs. *Intensive* (*b*) graph for no-retransmission

Figs 6.1 illustrate the relationship between *degree (k)* and the range of *intensive (b)* with the supremum and infimum that satisfy Eq. (5.8),(5.12) and (5.17) respectively for the case BD,DB and IM Update Rules respectively for case no-retransmission.

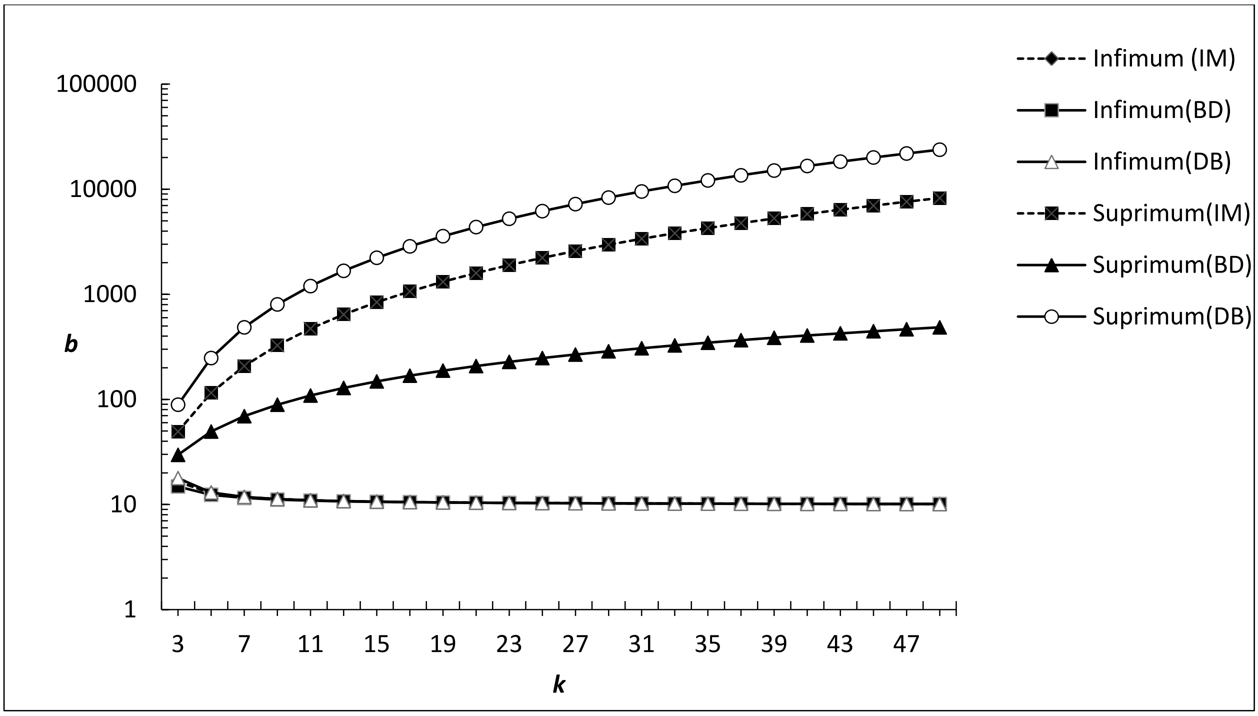


Fig.6. 1 *Degree (k)* vs. *Intensive (b)* graph for three update rules for case no retransmission

- **Observation**:

  When we compare the intensive with respect to degree, infimum shows the same characteristics for three different update rules and the values are much nearer to each other. So all infimum is at the same line .But the supremum shows the different values. For the same values of *degree (k)*, the supremum of DB has the highest value of intensive (*b)* and the supremum of BD has the lowest value of intensive (*b)* which is shown in the figure 6.1 above for *degree (k)* vs. *intensive (b)* graph for three update rules.

- **Discussion**:

  This is because the interaction among nodes is less in DB and more in BD. So message ferry needs to supply more *b* in DB so the range of *b* is widen in DB but as the node are more interactive in BD ,the message ferry need to supply only a small amount of energy that is *b* and nodes are willing be aggregators more so the range is small is BD.

  But we will not consider the supremum. Supremum can be higher as much as it can. Here we will consider only the infimum value which are almost same and nearer.

- **Better Update Rule for Novel Strategy Selection**:

  We cannot take any decision or select better update rule because the infimum shows the same value in three cases. So, all three update rules shows the same characteristics.

6.1.2 *Degree* (*k*) vs. *Modifier matrix (m)* graph for variable *intensive (b)*

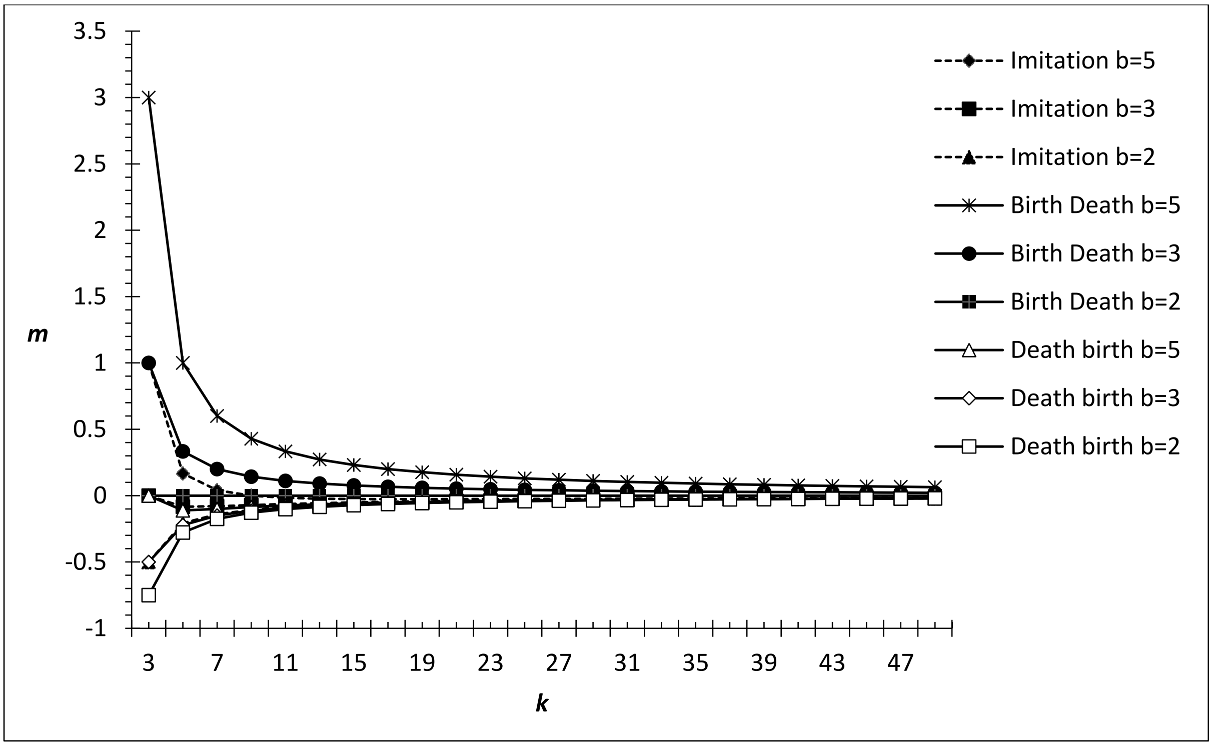


Fig.6. 2 *Degree (k)* vs. *Modifier Matrix (m)* graph for three update rules for variable *b* for no-retransmission

- **Observation**:

  Fig.6.2 shows the relationship between *degree* (*k*) *and modifier matrix* (*m*).For the same value of *k* for variable *b*, BD has the highest value of *m* and DB has the lowest value of *m* which is shown in figure 6.2. Here we take the value of *b*= 5, 3 and 2 respectively within the range. Here the range of *modifier matrix* (*m*) is varying for three cases of no-retransmission. For the case of IM update rule, the range is -0.8 to 1, for BD update rule the range is -0.5 to 3 and for DB update rule, the modifier matrix range is -0.8 to 0. Here the graph follows the Eq. (5.5). After a certain time, after *k>25*, all graph converges to zero that means *m* converges to zero.

- **Discussion:**

  This is because, as the BD update Rules requires smaller *b* and nodes are more willingly preferred to be aggregators than senders, the *modifier matrix* converges largest value for BD update rule. This can be said when *k<25*. After *k>25*, all graphs converges to zero.

- **Better Update Rule for Novel Strategy Selection:**

  Among three update rules, we can select BD update rule is better when we vary *intensive* (*b)* and plot *degree* (*k)* vs. *modifier matrix* (*m)* graph for *k*<25. After *k>25*, we can’t fix any update rule is as better to select the strategy in this case as all curves converges to zero.

6.1.3 *Degree* (*k)* vs. *Ratio of Aggregators* (*x*)* graph

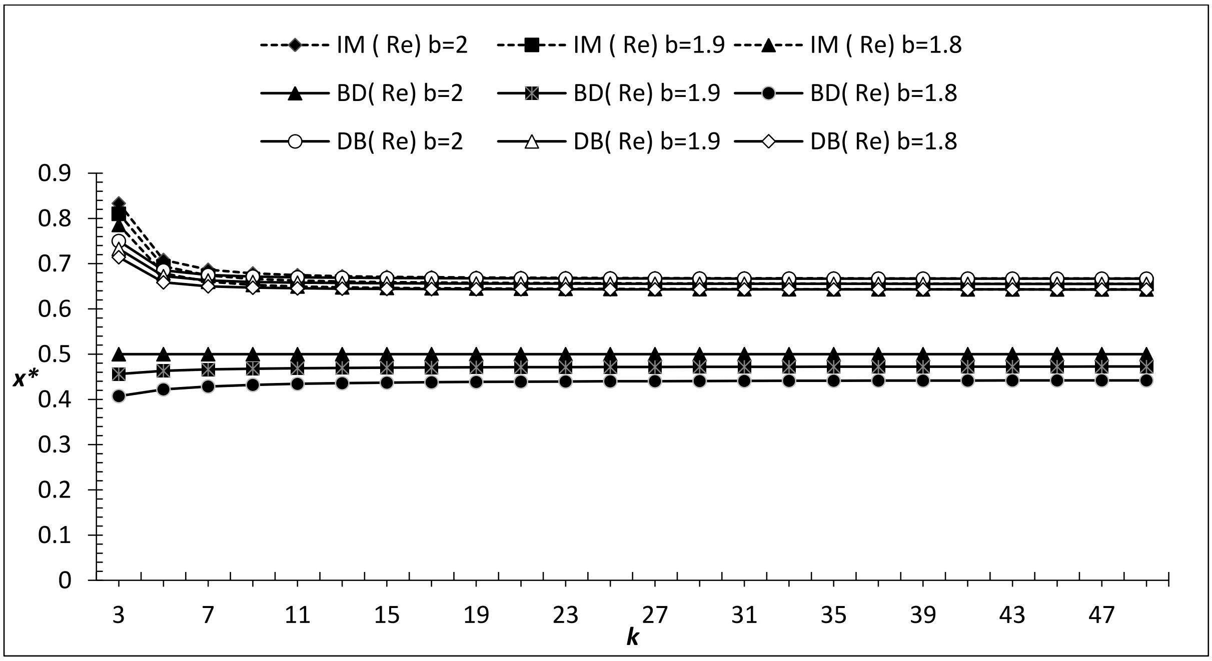


Fig.6. 3 *Degree (k)* vs. *Ratio of Aggregators (x*)* graph for three update rules for variable *b* for the case no retransmission

- **Observation**:

  In this figure, we show how *ratio of aggregators* vary when we plot *degree* (*k*) vs. *ratio of aggregators* (*x**) curve for variable of *intensive* (*b*) in Fig.6.3.The number of the *ratio of aggregators* are within the range between 0 and 1. For the same value of *k* for variable *b*, IM update rule has the highest value of *ratio of aggregators (x*)* for the lowest value of *k* (*k<10*) and BD has the lowest value of *ratio of aggregators (x*).*When *k>10,* IM and DB has the nearer values of the *ratio of aggregators(x*).*

- **Discussion:**

  For a specific *b*, *x** does not change when *k* becomes large. This is because *m* converges to zero with an increase of *k* as shown in Fig.6.3. On the other hand, if *k* is less than 10, *x** shows different characteristics, depending on *b*. The smaller *b* is, the lower *x* is, and vice versa.

  If we change the value of *b* from lower to higher, in IM update rule, the nodes have the highest intensity to be aggregators than senders and nodes are more co-operative in nature. So the *ratio of aggregators (x*)* are getting more than any other update rules for IM when k is small (*k>10*). After (*k >10*), IM and DB have the nearest values of *ratio of aggregators* (*x**).

  But in our case, it is better to get less number of *ratio of aggregators (x*)* which is given by update rule BD.

- **Better Update Rule for Novel Strategy Selection**:

  We can select BD update rule is for selection of novel strategy when we plot *degree (k)* vs. *ratio of aggregator (x*)* as it is giving the less number of *ratio of aggregators (x*)*

6.1.4 *Degree (k)* vs. *Intensive (b)* for variable ratio of aggregators (*x**)

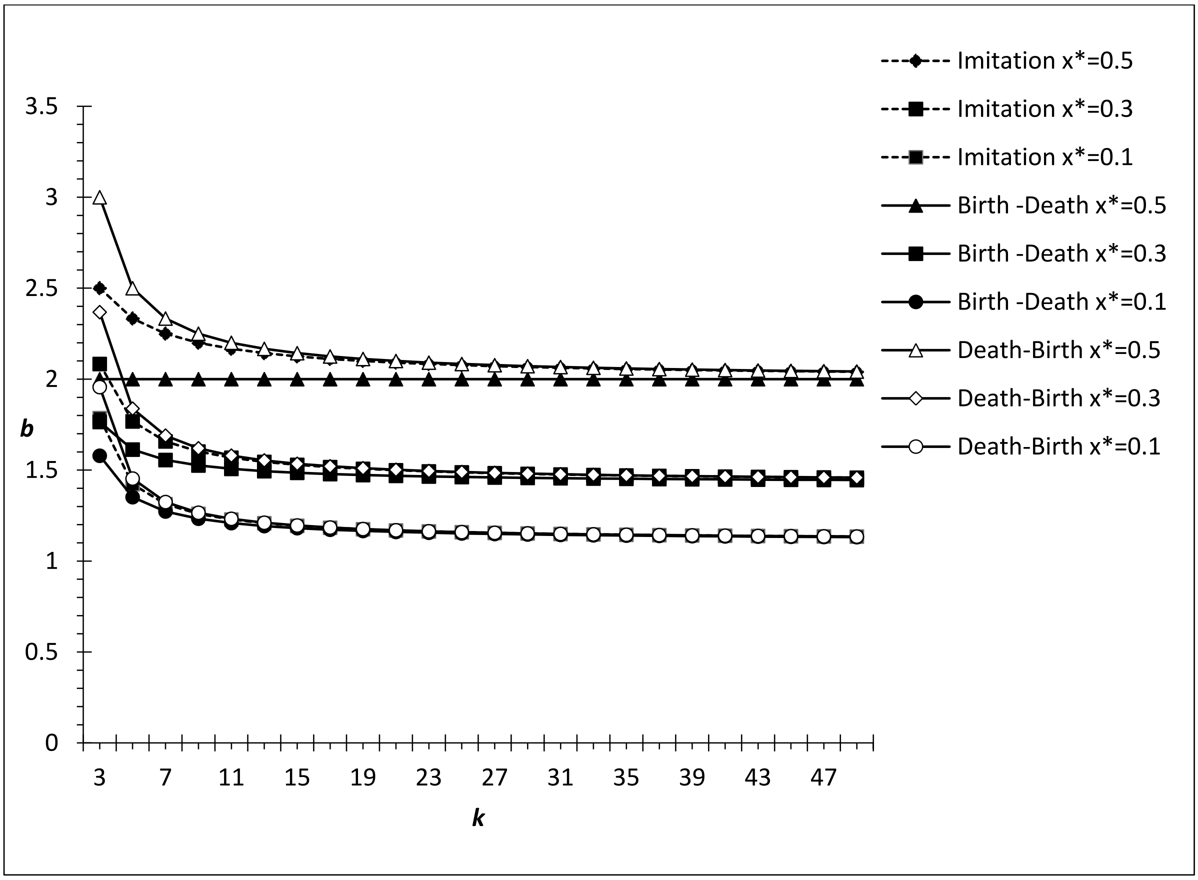


Fig.6. 4 *Degree (k)* vs. *Intensive (b)* graph for variable *ratio of aggregators (x*)* for the case no-retransmission

- **Observation**:

  This figure (Fig. 6.4) shows how *intensive* (*b*) varies with *degree* (*k*) when we vary *ratio of aggregators* (*x*). For the same value of *degree (k),* for variable ratio of *aggregators (x*)* (where *x**=0.5, 0.3 and 0.1) between three update rules DB has the highest value of *b* and the BD has the lowest value of *b* initially when degree is small (*k<20*). But when *degrees* are increasing (after *k>20*), all update rules nearly need same *b* at the *same ratio of aggregators.*

- **Discussion**:

  When *b* decreases, *the ratio of aggregators* also decreased for DB update rules more than any other update rules. So, if we change the *ratio of aggregators* from lower to higher, DB update rules require more *b* as nodes are more co-operative in nature.

Our objective is to get less number of *ratio of aggregators* (*x**) with lowest given intensive (*b*) which is fulfilled by update rule BD.

- **Better Update Rule for Novel Strategy Selection:**

  We can select update rule BD as better update rule when (k<20) when we plot *degree (k)* vs. *intensive (b)* for variable *ratio of aggregators (x*)* as message ferry have to supply less energy to nodes to become aggregators when number of *ratio of aggregators (x*)* is fixed. After *k>20*, we are unable to take any decision as three update rules have the same *intensive (b)* for the fixed ratio of aggregators.

6.1.5 *Degree* (*k)* vs. *intensive* (*b)* for variable *c* and *s*

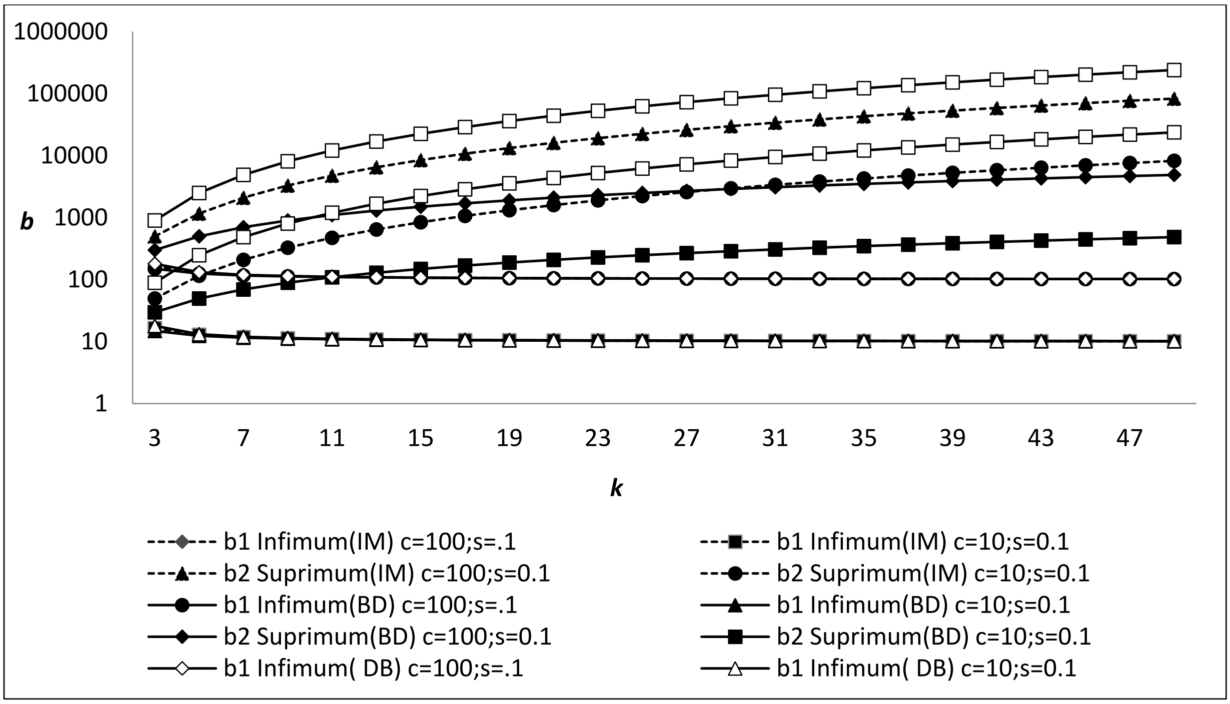


Fig.6. 5 *degree (k)* vs. *intensive (b)* graph for variable *c* and *s* for three cases for no retransmission

- **Observation:**

  This figure (Fig.6.5) show the relationship between *degree (k)* and the *intensive (b).*This figure is same as Fig 6.1 described in section 6.1.1. But the difference is that, in section 6.1.1, we assume *c*-*s*=1 for simplicity and in Fig.6.5 we vary *c* and *s* to see the impact of *c* and *s* on *intensive (b).*

  From the figure above (Fig.6.5), we can see that, if we increase the value of *c*, the graph shifts upwards for three cases as well. Now for the same value of *k* for variable *c*, the infimum shows the same characteristics but the supremum shows the different characteristics. The infimum has almost the same value of three cases so the value of *b* doesn't change almost but for the case of supremum, for higher *c*, the supremum of DB has the largest value and the supremum of BD has the minimum value of *b*. Again, that increase of *s* decreases the supremum of *b*.

- **Discussion:**

  This is because when senders lose more energy, temptation *b* to become an aggregator can be smaller. Again when we increase the value of *c*, the graph shifts upwards, that directly indicates that $b>c$.

- **Better Update Rule for Novel Strategy Selection:**

  We cannot give any decision in this case also when we vary c and s and plot *degree (k)* vs. *intensive (b)* graph as all the infimum lies in the same line. So, all three update rules show the same characteristics.

6.2 Comparison for retransmission case

6.2.1 *Degree (k)* vs. *intensive* (*b*) graph for no-retransmission

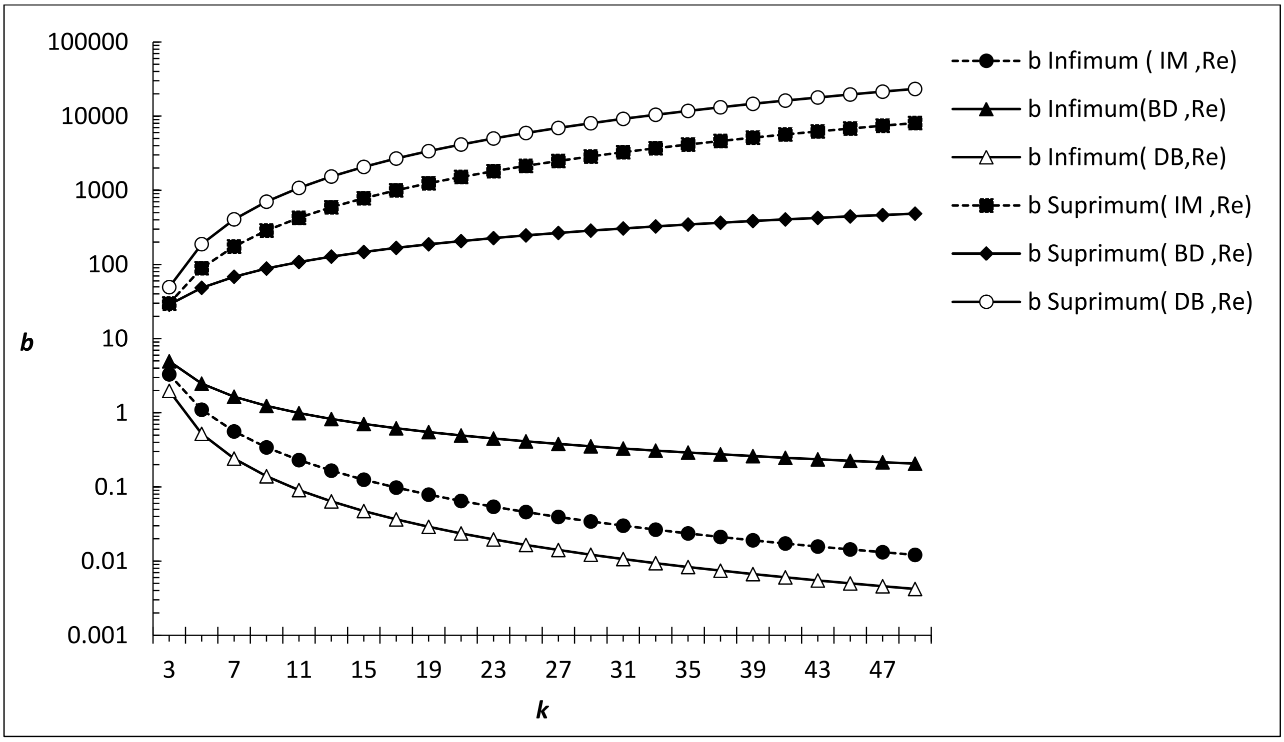


Fig.6. 6 *degree (k)* vs. *intensive (b)* graph for the case retransmission

- **Observation**:

  in Fig.6.6, we show the relationship between *intensive (b)* with *degree (k).* For the same value of *k*, DB supremum have the highest value of *b* and BD supremum has the lowest value of *b*.

  On the other hand, infimum is not constant like no-retransmission case. Here for the same value *k*, the range of *b* is lowest value of infimum for DB and highest value BD retransmission case.

- **Discussion**:

  This is because the interaction among nodes is less in DB and more in BD. So, message ferry need to supply more *b* in DB if nodes are willingly to be aggregators. So the range of *b* is widen in DB but as the node are more interactive in BD, the message ferry need to supply a only small amount of energy that is *b* to the nodes for becoming aggregators so the range is small is BD. That means the node prefer to be aggregators to avoid lowest pay off.

  If we consider the infimum, DB has the lowest value for infimum.

- **Better Update Rule for Novel Strategy Selection**:

  As infimum DB has the smallest value of *b*, we can select DB update rule as better update rule for selection of novel strategy when we plot *degree (k)* vs. *intensive (b)* as message ferry need to supply only a small amount of energy to node becoming an aggregator.

6.2.2 *Degree* (*k*) vs. *Modifier matrix* (*m*) graph for variable i*ntensive* (*b*)

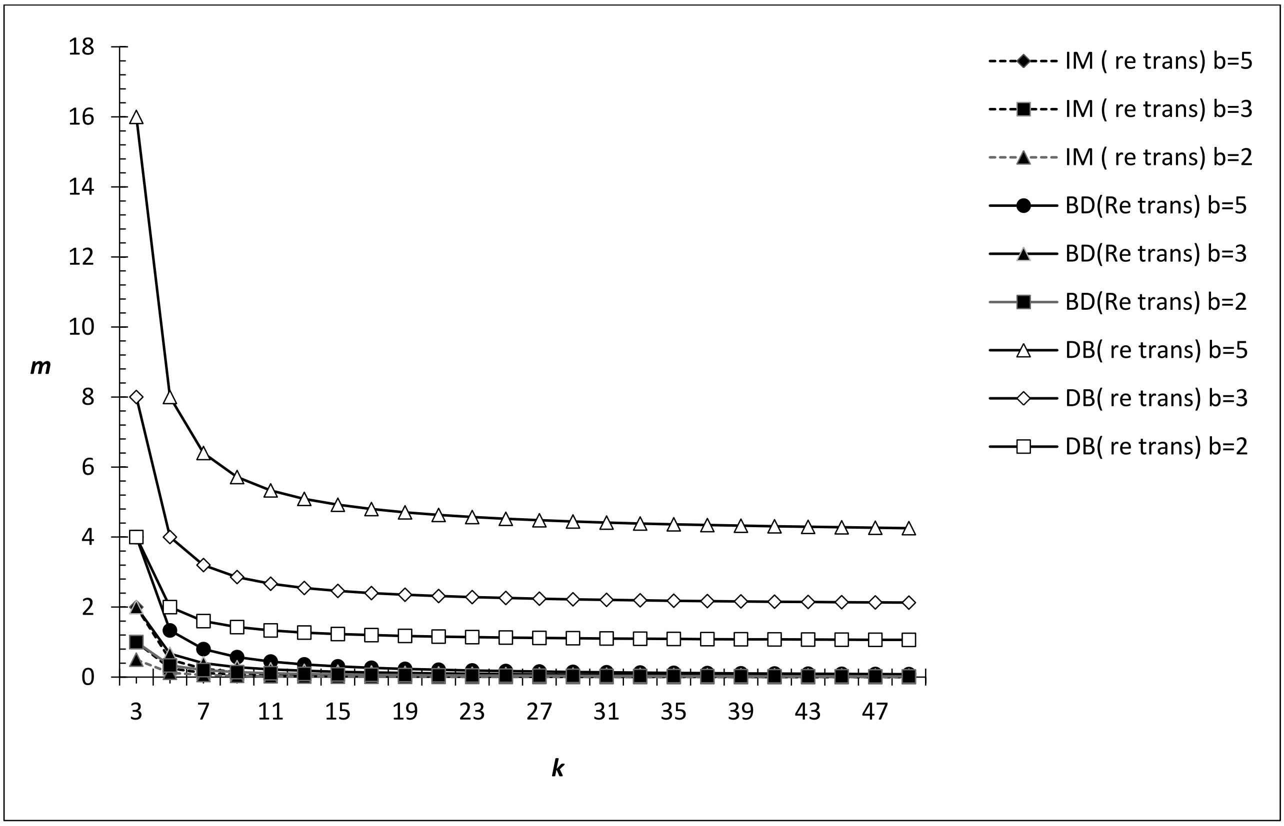


Fig.6. 7 *Degree (k)* vs. *Modifier matrix (m)* graph for variable *b* for the case retransmission

- **Observation**:

  This graph shows the relationship between *degree* (*k*) and *modifier matrix* (*m*). Here we take the values of b is 5, 3 and 2 respectively within the range .For the highest value of intensive (*b*), DB has the highest value of *modifier matrix* (*m*) varies with degree (*k)* and IM has the lowest value. Same thing happens for the lowest value of *m*. That means for the variable value of *b*, DB has the highest value of *m* varies with *k* and IM has the lowest value of *m*. This can be considered only when k is small and k<20.When the value of *k* is increasing, the value of m doesn't change so much.

- **Discussion:**

  As the quantity (*c-s*) is fixed in case of re-transmission, the *modifier matrix* only dependent on the value of *b*. As DB has the highest range of values for *b,* in *modifier matrix*, DB also have the highest value for *m*.

- **Better Update Rule for Novel Strategy Selection:**

  As DB update rule gives the highest values of *modifier matrix m* for *k<20,* we can consider DB as the better update rule for selection of novel strategy when we plot *Degree (k)* vs. *Modifier matrix (m).*

6.2.3 *Degree* (*k*) vs. *Ratio of Aggregators* (*x**) graph for variable *intensive* (*b*)

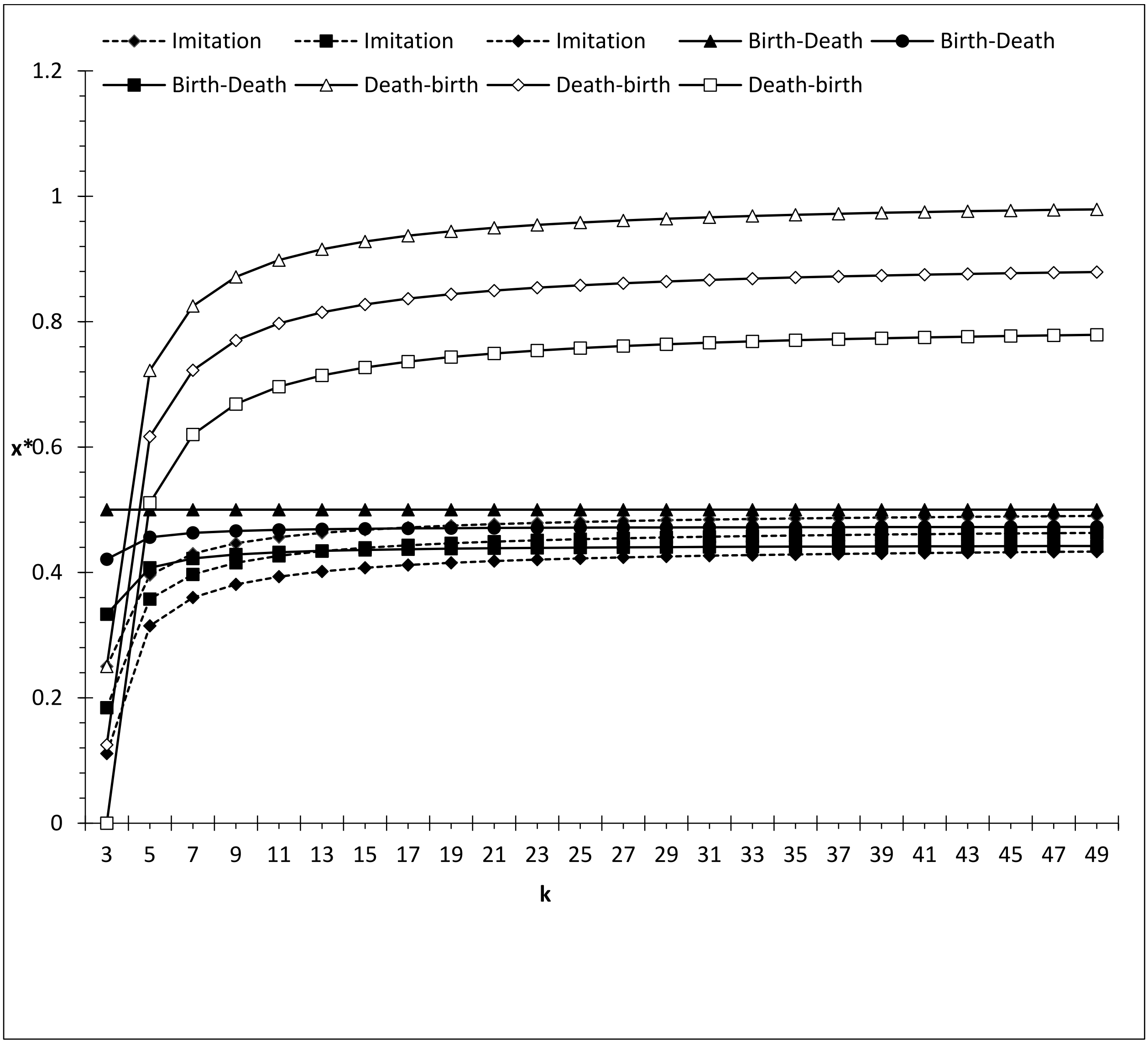


Fig.6. 8 *Degree (k)* vs. *Ratio of Aggregators (x*)* graph for variable *b* for the case retransmission

- **Observation**:

  This figure shows the relationship between *Degree (k)* and *Ratio of Aggregators* (*x*)* for variable *intensive* (*b*).The *ratio of aggregators (x*)* can be any value between 0 and 1.

  For the same value of *k*, for variable *b* (b=2, 1.9, and 1.8 respectively), the update rule BD has the highest values initially, and IM and DB has the lowest values, when *k* (*k*<3) is small. When *k* becomes large, the *Ratio of Aggregators* (*x*)* are also increasing drastically in DB update rule which is more than any other update rules. After a certain values of *k*, for a certain values of *b*, the number of the *ratio of aggregators* are (*x**) are not changing.

- **Discussion**:

  This is because when we increase the values of *b*, the nodes are more interactive in BD Update Rule and willingly to be aggregators more than being senders. So, Update Rule BD has the highest values of ratio of aggregators and DB and IM Update Rule has the lowest value for the *ratio of aggregators* for the same values of *b* when *k* is small (*k*<5).When *k* increases (*k*>5), the *ratio of aggregators* are increasing in update rule DB as nodes have temptation to become aggregators than senders.

- **Better Update Rule for Novel Strategy Selection**:

  We can say, initially update rule DB and IM are best here because, for the same amount of energy supplied by message ferry, we get small number of *ratio of aggregators* (*x**). When *k* increases, after *k*>5, only update rule IM is better as it gives least amount of the *ratio of aggregators* (*x**).

6.2.4 *Degree* (*k*) vs. *Intensive* (*b*) graph for variable *Ratio of Aggregators* (*x**)

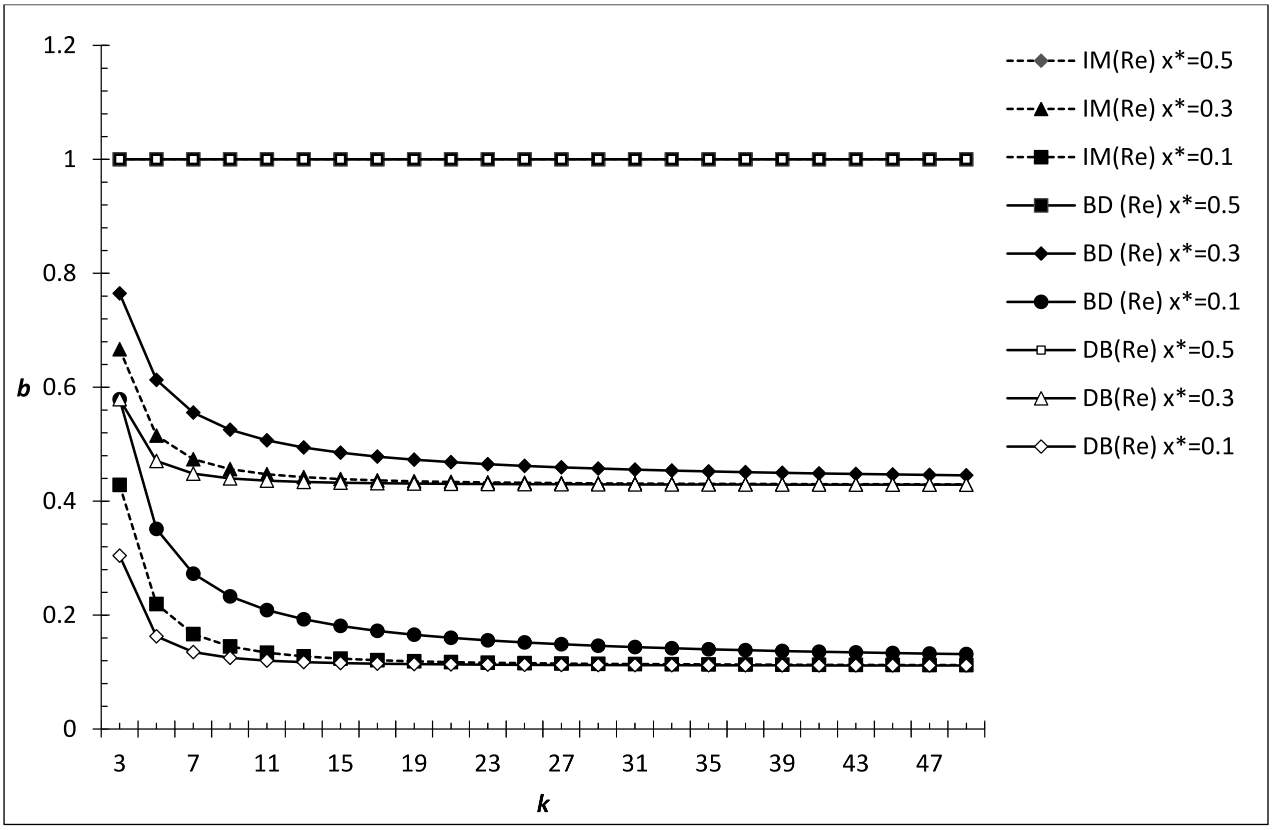


Fig.6. 9 *Degree (k)* vs. *Intensive (b)* graph for variable *x** for the case retransmission

- **Observation:**

  This graph shows the relationship between *Degree* (*k*) and *Intensive* (*b*) when we vary *Ratio of Aggregators* (*x**). We take the values of *x** are 0.5, 0.3 and 0.1 respectively. For the highest value of *x**, for the same value of *k* , three update rules (IM,BD and DB) has the same values ok *b* that is 1 and the values does not change.

  Again if we decrease the values of *x** to 0.3, BD has the highest values of *b* and DB has the lowest values of *b* and the values are decreasing. After a certain values of *k*, if we increase the values of *k*, the value of *b* does not change *(k>25)*. Same thing happens if we decrease the values of *x** again.

- **Discussion**:

  This is because, for the highest values of aggregators, all three update rules consumes nearer same amount of *b* supplied by message ferry. But when aggregator decreased from the higher value to lower values gradually, the energy supplied by message ferry also decreased but this decrement is faster for the case of DB.

- **Better Update Rule for Novel Strategy Selection:**

  We can select update rule DB is better as in this rule for the same level of *b*, less number of *ratio of aggregators* (*x**) we are getting, when *k<25*. When *k* increases to higher level, we can't take any decision as for the same level of *ratio of aggregators*, we are getting same level of *intensive* (*b*).

6.2.5 *Degree* (*k*) vs. *Intensive* (*b*) for variable *c* and *s*

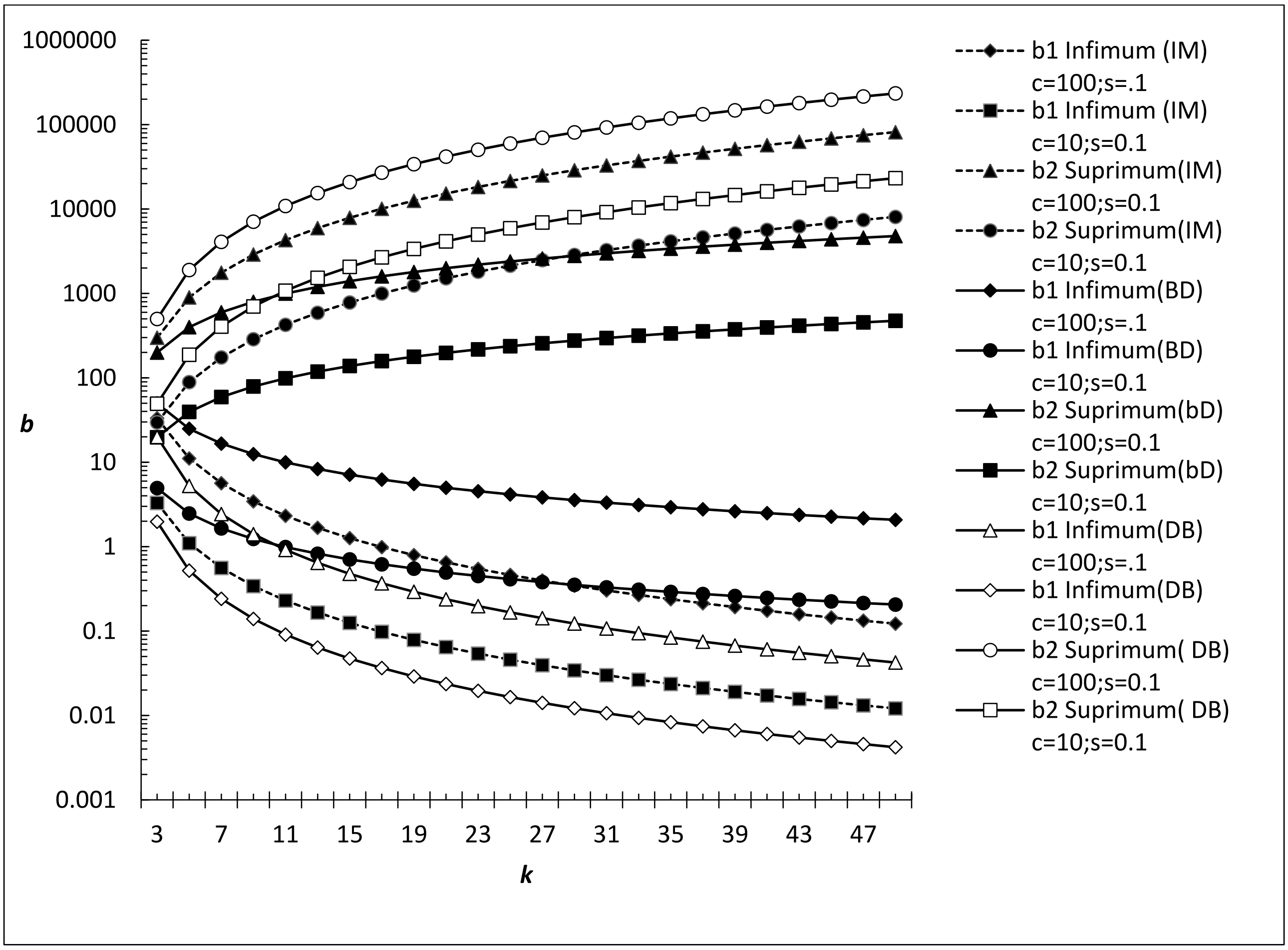


Fig.6. 10 *Degree (k)* vs. *Intensive graph (b)* for variable *c* and *s* for the case retransmission

- **Observation**:

  This figure shows the relationship between degree (*k*) and intensive (*b*) we vary *c* and *s*. This figure is same as Fig.6.5 described in section 6.2.1.But the difference is, that case we assumed *c*-*s*=1 and this case we vary *c* and *s* to observe the impact of *c* and *s* on the graph.

For the same value of *k*, supremum DB have the highest value of *b* and BD supremum has the lowest value of *b*. On the other hand, infimum is not constant like no-retransmission case. Here for the same value k, *b* has the lowest value of infimum for DB and highest value BD retransmission case. If we decrease the value of *c*, the supremum decreases and if we increase the value of s, the supremum shifts upward.

- **Discussion:**

This is because the interaction among nodes is less in DB and more in BD. So message ferry have to supply more *b* in DB so the range of *b* is widen in DB but as the node are more interactive in BD ,the message ferry need to supply a only small amount of energy that is b to the nodes to become aggregators so the range is small is BD. That means the node prefer to be aggregators to avoid lowest pay off. 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.

Again, we increase the values of *c*, the graph shifts upwards which directly indicates that $b>c$ and $b-c$ should be positive.

Again, if we consider the infimum, infimum DB has the lowest values of b with respect to others. It also happens when *degree* (*k*) is increasing.

- **Better Update Rule for Novel Strategy Selection:**

As DB update rule needs lowest *b* to nodes to become aggregators when increase in degrees (*k*) at infimum, we can choose DB as the best update rule in this aspect.

6.3 Summary of the numerical Analysis

So we can summarize the above scenario by following:

6.3.1 For the case of no-retransmission

Comparison	**Best Update rule for Novel Strategy for selection to be an aggregator**
Degree (k) **vs.** ***Intensive (b)*** **graph**	All three update rules show the same characteristics
Degree (k) **vs.** ***Modifier Matrix (m)*** **graph for variable** ***Intensive*** **(*b*)**	Birth-Death when $k<25$, after $k>25$,all three update rules show the same characteristics.
Degree (k) **vs.** ***Ratio of Aggregator (x*)*** **graph for variable** ***Intensive (b)***	Birth-Death
Degree(k) **vs.** ***Intensive (b)*** **graph for variable** ***x****	Birth-Death when $k<20$,after $k>20$, all three update rules show the same characteristics
degree vs. ***intensive graph*** **for variable** ***c*** **and** ***s***	All three update rules show the same characteristics

6.3.2 For the case of retransmission

Comparison	**Best Update rule for Novel Strategy for selection to be an aggregator**
Degree (k) **vs.** ***Intensive (b)*** **graph**	Death -Birth
Degree (k) **vs.** ***Modifier Matrix*** **graph for variable** ***Intensive (b)***	Death-Birth for $k<20$
Degree (k) **vs.** ***Ratio of Aggregators (x*)*** **graph for variable** ***Intensive (b)***	Death-Birth initially when $k<5$,after $k>5$, Imitation
Degree (k) **vs.** ***Intensive (x*)*** **graph for variable Ratio of Aggregators (*****x*)***	Death-Birth when $k<25$,after $k>25$, all three update rules show the same characteristics for smaller x^*
Degree (k) **vs.** ***Intensive (b) graph*** **for variable** ***c*** **and** ***s***	Death-Birth

Discussion: We have four independent variables, *b, c, s* and *k,* which affect *x.* We plot the variables with respect to *k* in the graphs shown above. Here, we can see, for a specific plotting, one specific update rule is selected as Novel or best update rule.

When degree (*k*) is increasing with respect to intensive (*b*), for the update rule at which *b* is smaller among three update rule, we select that update rule as better. Again, if three update rules give the same level of *intensive (b),* we cannot choose any update rule as better for selection of strategy.

Again, when we want to select the ratio of aggregators, then we introduce variable *x*.* Normally it is better to obtain a small number of *ratio of aggregators* (*x*).*So, when we consider *ratio of aggregators* (*x*)* with respect to *intensive* (*b*), we consider that update rule as novel which gives less *ratio of aggregators* (*x*)* at the same amount of *intensive* (*b)*

Again when we consider the modifier matrix (*m*) with respect to *degree* (*k*), we consider the fact that, the update rule which give the highest values of *modifier matrix (m),* we consider that update rule is as novel. Highest values of *modifier matrix* means nodes are more interactive and willingly want to be an aggregator with the lowest values of intensive given.

Finally, when we vary *c* and *s* at the same time, we want to observe the impact of *c* and *s* on intensive *(b),* when *intensive* (*b*) is plotted against *degree* (*k*). After observing the impact, we consider the better update rule for selecting the novel strategy.

After above discussion we can select Birth-Death Update Rule suits best for the case of no retransmission and for the case of retransmission, Death-Birth Update Rule when degree is small and Imitation Update Rule for higher degrees suits best as these update rules give the best result that is least number of ratio of aggregators with lessa energy supplied by message ferry.

6.4 Conclusion

In this chapter we described our numerical analysis for the proposed mathematical model for the case of no retransmission and retransmission respectively which are discussed in previous chapter. We have shown observation for each graph with brief discussion and after that we select the best update rule for selecting the strategy. We found that,

Chapter 7

Conclusion

This thesis 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, the Self-Organized Data Aggregation scheme in such an environment is discussed. In this scheme, the selection of aggregators is conducted through decentralized processes with the help of strategic decisions of evolutionary game theory.

Chapter 2 addressed about Ferry Assisted Delay Tolerant Network (DTN). It is known that DTN is a unique technique that enables communication and data transfer in situations where traditional networks are obsolete, i.e., where, physical end-to-end connectivity is not present. By using DTN technique, we can practically establish connectivity within a closed remote boundary using wireless equipped devices, where devices connect each other using Ad-Hoc networking.

Chapter 3 focused on 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.

Chapter 4 briefly describes about Game Theory, Nash Equilibrium, Applications of game theory and Nash Equilibrium, Evolutionary Game Theory, Replicator Equation on Graph and Modifier Matrix. Self-Organized Data Aggregation techniques can be solved by the help of Evolutionary Game theory in order to take account of the inherent selfishness of the nodes for saving their own battery life which is described in this chapter. In Evolutionary Dynamics, Replicator Equation is one of the most fundamental equations. Note that, Replicator Equation is used for infinite number of players and Replicator Equation on graph is for finite number of players .That’s why, in our thesis, we use Replicator Equation on Graph.

Chapter 5 basically focused on proposed mathematical model for our thesis. For this, abstract payoff matrix, pay off matrix for the case no retransmission and retransmission case was discussed respectively. We introduced valid parameters for system stability, solvability and successful bundle re transmission and no retransmission, energy consumption for two cases, three update rules of evolutionary game theory and mathematical calculations for no retransmission and retransmission respectively and summary for three update rules. Note that, the number of aggregators can be controlled to a desired value by adjusting the energy that the message ferry supplies to the aggregators.

Chapter 6 describes the numerical analysis for the mathematical model that was described in chapter 5. Here Degree vs. Intensive curve, Degree vs. Modifier Matrix curve by varying Intensive, Degree vs. Ratio of Aggregators curve by varying Intensive, Degree vs. Intensive curve when Ratio of Aggregators are varying and finally Degree vs. Intensive curve when energy consumed by aggregators and senders are varying for the case of no retransmission and retransmission respectively are plotted. We include observation, discussion and finally better update rule selection part for particular plotting for two cases respectively.

In this thesis, we wanted to select the Novel Strategy for Ferry Assisted Multi Cluster DTNs for selection of aggregators. For this we have introduced some mathematical calculations, perform numerical analysis and analysis them. After that, we try to give a brief discussion which update rule is better in which aspect. Finally we have select better update rule for the case no retransmission and retransmission respectively. For the case of no retransmission we have selected Birth-Death Update Rule as better update rule for strategy selection of the aggregators and for the case of retransmission we have selected Death-Birth Update Rule when degree is small and Imitation Update Rule is as better when degree is higher as better update rules.

However, for the shortage of time, we cannot perform the simulation of our numerical analysis for proposed mathematical model. These numerical analysis can be verified over simulation and more proper and perfect analysis can be given so, this thesis can be extended further in future by doing simulation the numerical analysis of proposed mathematical model over Net logo software and verified the numerical analysis with the simulated result.

Appendix A

Birth Death-Update Rule (BD) for the case of retransmission

First, we will derive the replicator equation on graphs for the case retransmission as described in Table 5.2.

Let *x* denote the ratio of the number of aggregators to the total number of cluster members. Note that $1 - x$ represents the ratio of the number of senders.

The Modifier Matrix for the case of BD is given by [46-48]

$$m_{ij(BD)} = \frac{a_{ii+}a_{ij-}a_{ji-a_{jj}}}{k-2} \qquad \text{(A.1)}$$

The expected payoff (fitness) $f_{1(DB,Re)}$ and $f_{2(DB,Re)}$ of aggregators and senders respectively are obtained by substituting the value of Table 5.2 into (Eq. 4.2), which are as

$$f_{1(DB,Re)} = x(-c) + (1-x)(b-c), \qquad \text{(A.2)}$$

$$f_{2(DB,Re)} = x(c-s) - c$$

respectively.

The local competition is given by putting the value of Table 5.2 into Eq. (4.6)

$$g_{1(DB,Re)} = (1-x)m, \qquad \text{(A.3)}$$

$$g_{2(BD,Re)} = -xm$$

After putting the value of Table 5.2, the modifier matrix from Eq. (C.1) is given by

$$m_{ij(BD,Re)} = \frac{b-c+s}{k-2} \qquad \text{(A.4)}$$

The average pay off can be obtained using Eq. (4.8) and Table 5.2 is given by

$$\begin{aligned} \Phi_{(BD,Re)} &= x_1 \left(f_{1(BD,Re)} + g_{1\,(BD,Re)}\right) + x_2 \left(f_{2\,(BD,Re} + g_{2(BD,Re)} \right) \\ &= (b+c-s)(1-x)x - x \qquad \text{(A.5)} \end{aligned}$$

where, $x_1 = x$, denotes the ratio of Aggregators and $x_2 = 1 - x$, denotes the ratio of senders.

Finally the replicator equation on graph for being an aggregator is obtained

$$\begin{aligned} \dot{x_1} &= x_1[f_{1(DB,Re)} + g_{1(DB,Re)} - \Phi_{(BD,Re)}] \\ &= x(1-x)[b + \frac{b-(c-s)}{k-2} - bx - x(c-s)] \qquad \text{(A.6)} \end{aligned}$$

Substituting $\dot{x_1}$=0, there exists three equilibria $x = 0{,}1$ and

$$x_{2}{}^{*}_{(BD,Re)} = \frac{b(k-2) + b - (c-s)}{b(k-2) + (k-2)(c-s)} \qquad \text{(A.7)}$$

Note, equilibria is feasible if 0<x<1, i.e,

$$\frac{1}{k-1} < \frac{b}{c-s} < k-1 \tag{A.8}$$

holds, where we use $c - s > 0$. We also have for all $k > 2$, $0 < 1/(k-1) < k-1$. As a result, for any c, s, and k, there exists $b > 0$. Thus the equilibria is controllable and it can be shown to be stable.

Appendix B

Death-Birth Update Rule (DB) for the case of no-retransmission

First, we derive the replicator equation on graphs [8] for case no retransmission as described in Table 5.1.

Let x denote the ratio of the number of aggregators to the total number of cluster members. Note that $1 - x$ represents the ratio of the number of senders.

The modifier matrix [46-48] for the case of DB update rule is

$$m_{ij(DB)} = \frac{(k+1)a_{ii+}a_{ij-}a_{ji} - (k+1)a_{jj}}{(k+1)(k-2)} \tag{B.1}$$

The expected payoff (fitness) $f_{1(DB)}$ and $f_{2(DB)}$ of aggregators and senders respectively are given by substituting the value of Table 5.1 into Eq. 4.2

$$f_{1(DB)} = x(-c) + (1-x)(b-c), \tag{B.2}$$

$$f_{2(DB)} = -s$$

respectively.

The expected pay-off for local competition is given by using the values of Table 5.1 into Eq. (4.6)

$$g_{1(DB)} = (1-x)m, \tag{B.3}$$
$$g_{2(DB)} = -xm$$

respectively.

By substituting the value of Table 5.1 into Eq. (A.1) , the modifier matrix is given by

$$m_{ij(DB)} = \frac{(k+1)(s-c) + (c-s-b)}{(k-2)(k+1)} \tag{B.4}$$

The average pay-off cab be obtained by using Eq. (4.8)

$$\begin{aligned} \Phi_{(DB)} &= x_1\ (f_{1\,(DB)}\ + g_{1(DB)}\ ) + x_2\ (f_{2(DB)}\ + g_{2(DB)}\ ) \\ &= (bx - s)(1-x) - cx \end{aligned} \tag{B.5}$$

where, $x_1 = x$, denotes the ratio of Aggregators and $x_2 = 1 - x$, denotes the ratio of senders.

Finally the replicator equation on graph for being an aggregator is obtained by

$$\begin{aligned} \dot{x_1} &= x_1(f_{1(DB} + g_{1(DB)} - \Phi_{(DB)}) \\ &= x(1-x)[\frac{b\{(k^2 - k - 2) + 1\} - k^2(c-s)}{(k+1)(k-2)}] \end{aligned} \tag{B.6}$$

Substituting $\dot{x}_1$=0, there exists three equilibria $x = 0,1$ and

$$x_{1}{}^{*}_{(DB)} = \frac{b(k^2 \ - k - 2) - k^2(c - s) + b}{b(k^2 - k - 2)} \tag{B.7}$$

Note,equilibria is feasible if 0<x<1, i.e,

$$\frac{k^2}{k^2 - k - 1} < \frac{b}{c - s} < k^2 \tag{B.8}$$

holds, where we use $c - s$ >0. We also have for all k >2, 0 <$\frac{k^2}{k^2 - k - 1}$<k^2. As a result, for any c, s, and k, there exists b >0. Thus the equilibria is controllable and it can be shown to be stable.

Appendix C

Death-Birth Update Rule (DB) for the case of retransmission

First, we derive the replicator equation on graphs as described in Table 5.2 for the case of retransmission.

Let *x* denote the ratio of the number of aggregators to the total number of cluster members. Note that $1 - x$ represents the ratio of the number of senders.

The modifier matrix [46-48] for the case of DB update rule is

$$m_{ij(DB)} = \frac{(k+1)a_{ii+}a_{ij-}a_{ji} - (k+1)a_{jj}}{(k+1)(k-2)} \tag{C.1}$$

The expected payoff (fitness) $f_{1(DB,Re)}$ and $f_{2(DB,Re)}$ of aggregators and senders respectively are given by substituting the value Table 5.2 into Eq. (4.2) are

$$\begin{aligned} f_{1(DB,Re)} &= x(-c) + (1-x)(b-c), \\ f_{2(DB,Re)} &= x(c-s) - c \end{aligned} \tag{C.2}$$

respectively.

The expected pay-off for local competition is obtained by substituting the values of Table 5.2 into Eq.(4.6) are

$$\begin{aligned} g_{1(DB,Re)} &= (1-x)m, \\ g_{2(DB,Re)} &= -xm \end{aligned} \tag{C.3}$$

respectively.

The modifier matrix is given by using the values of Table 5.2 into Eq. (C.1) is

$$m_{ij(DB,Re)} = \frac{b-c+s}{(k-2)(k+1)} \tag{C.4}$$

The average pay off can be obtained using Eq. (4.3) is given by

$$\begin{aligned} \Phi_{(DB,Re)} &= x_1\,(f_{1(DB,Re)} + g_{1(DB,Re)}) + x_2\,(f_{2(DB,Re)} + g_{2(DB,Re)}) \\ &= (b+c-s)(1-x)x - c \end{aligned} \tag{C.5}$$

where, $x_1 = x$, denotes the ratio of Aggregators and $x_2 = 1 - x$, denotes the ratio of senders.

Finally the replicator equation on graph for being an aggregator is obtained by

$$\begin{aligned} \dot{x_1} &= x_1(f_{1(DB,Re)} + g_{1(DB,Re)} - \Phi_{(DB,Re)}) \\ &= x(1-x)[b + \frac{b-(c-s)}{(k-2)(k+1)} - x(b+c-s)] \end{aligned} \tag{C.6}$$

Substituting $\dot{x_1}$=0, there exists three equilibria $x = 0{,}1$ and

$$x_{1(DB,Re)}^{*} = \frac{b(k-1)(k-2) + b - (c-s)}{(k-2)(k+1)(b+c-s)} \tag{C.7}$$

Note, equilibria is feasible if 0<x<1, i.e,

$$\frac{1}{k^2 - k - 1} < \frac{b}{c-s} < k^2 - k - 1 \tag{C.8}$$

holds, where we use $c - s$ >0. We also have for all k >2, 0 <$\frac{1}{k^2-k-1}$<$k^2 - k - 1$. As a result, for any c, s, and k, there exists b >0. Thus the equilibria is controllable and it can be shown to be stable.

Appendix D

Imitation Update Rule (IM) for no-retransmission case

First, we derive the replicator equation on graphs [8] for case no retransmission as described in Table 5.1.

Let *x* denote the ratio of the number of aggregators to the total number of cluster members. Note that $1 - x$ represents the ratio of the number of senders.

The modifier matrix [46-48] for the case of IM update rule is

$$m_{ij(IM)} = \frac{(k+3)\, a_{ii} + 3a_{ij} - 3a_{ji} + (k+3)a_{jj}}{(k-2)(k+3)} \tag{D.1}$$

The expected payoff (fitness) $f_{1(IM)}$ and $f_{2(IM)}$ of aggregators and senders are given by substituting the values Table 5.1 into Eq. (4.2), that's are

$$f_{1(IM)} = x(-c) + (1-x)(b-c), \tag{D.2}$$

$$f_{2(IM)} = -s$$

respectively.

The local competition is given by substituting the values of Table 5.1 into Eq. 4.6, which are

$$g_{1(IM)} = (1-x)m, \tag{D.3}$$

$$g_{2(IM)} = -xm$$

respectively.

By substituting the values of Table 5.1 into Eq. (B.1), The modifier matrix is given by

$$m_{ij(IM)} = \frac{3b - (c-s)(k+6)}{(k-2)(k+3)} \tag{D.4}$$

The average pay-off cab be obtained by using Eq. (4.8)

$$\Phi_{IM} = x_1\, (f_{1(IM)} + g_{1(IM)}) + x_2\, (f_{2\,(IM)} + g_{2(IM)}) \tag{D.5}$$

$$= (bx - s)(1-x) - cx$$

where, $x_1 = x$, denotes the ratio of Aggregators and $x_2 = 1 - x$, denotes the ratio of senders.

Finally, the replicator equation on graph for being an aggregator is obtained by

$$\dot{x_1} = x_1(f_{1(IM)} + g_{1(IM)} - \Phi)$$

$$= x(1-x)[\frac{b(k^2+k-3) - (k^2+2k)(c-s)}{(k+3)(k-2)} - bx] \tag{D.6}$$

Substituting $\dot{x_1}$=0, there exists three equilibria $x = 0,1$ and

$$x_{1}{}^{*}_{(IM)} = \frac{b(k^2 \ + k - 3) - (k^2 + 2k)(c - s)}{b(k + 3)(k - 2)} \qquad \text{(D.7)}$$

Note, equilibria is feasible if 0<x<1, i.e,

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

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

Appendix E

Imitation Update Rule (IM) for the case of retransmission

First, we derive the replicator equation on graphs as described in Table 5.2.

Let *x* denote the ratio of the number of aggregators to the total number of cluster members. Note that $1 - x$ represents the ratio of the number of senders.

The modifier matrix [46-48] for the case of IM update rule is

$$m_{ij(IM)} = \frac{(k+3)\, a_{ii} + 3a_{ij} - 3a_{ji} + (k+3)a_{jj}}{(k-2)(k+3)} \tag{E.1}$$

By substituting the values of Table 5.2 into Eq.(4.2) ,the expected payoff (fitness) $f_{1(IM,Re)}$ and $f_{2(IM,Re)}$ of aggregators and senders are given by

$$f_{1(IM\,,Re)} = x(-c) + (1-x)(b-c), \tag{E.2}$$

$$f_{2(IM\,,Re)} = x(c-s) - c$$

respectively.

The expected pay-off for local competition is given by using Table 5.2 into Eq. (4.6) which are

$$g_{1(IM\,,Re)} = (1-x)m, \tag{E.3}$$

$$g_{2(IM,Re)} = xm$$

respectively.

The modifier matrix from Eq. (E.1) and Table 5.2 is given by

$$m_{ij(IM\ retrans)} = \frac{3(-b+c-s)}{(k-2)(k+3)} \tag{E.4}$$

The average pay off can be obtained using equation 7 is given by

$$\begin{aligned} \Phi_{(IM,Re)} &= x_1\ (f_{1(IM,Re)} + g_{1(IM,Re)}) + x_2\ (f_{2(IM,Re)} + g_{2(IM,Re)}) \\ &= bx + x(1-x)c - (2m-s) - x^2(b-c) \end{aligned} \tag{E.5}$$

where, $x_1 = x$, denotes the ratio of Aggregators and $x_2 = 1 - x$, denotes the ratio of senders.

Finally the replicator equation on graph for being an aggregator is obtained by

$$\begin{aligned} \dot{x_1} &= x_1(f_{1(IM,Re)} + g_{1(IM,Re)} - \Phi_{(IM,Re)}) \\ &= x(1-x)\left[\frac{b(k-1)(k-2) + b - (c-s)}{(k-2)(k+1)(b+c-s)}\right] \end{aligned} \tag{E.6}$$

Substituting $\dot{x_1}$=0, there exists three equilibria $x = 0{,}1$ and

$$x_{2}{}^{*}_{(IM,Re)} = \frac{b(k-1)(k-2)+b-(c-s)}{(k-2)(k+1)(b+c-s)} \tag{E.7}$$

Note, equilibria is feasible if 0<x<1, i.e,

$$\frac{3}{k^2+k-3} < \frac{b}{c-s} < \frac{k^2+k-3}{3} \tag{E.8}$$

Holds, where we use $c - s > 0$. We also have for all $k > 2$, $0 < \frac{3}{k^2+k-3} < \frac{k^2+k-3}{3}$. As a result, for any *c, s,* and *k,* there exists $b > 0$. Thus the equilibria is controllable and it can be shown to be stable.

References

1. K. Habibul Kabir, Masahiro Sasabe, and Tetsuya Takine, “Self-Organized Data Aggregation among Selfish Nodes in an Isolated Cluster,” in Proceedings of the 1stWorkshop on Bio-inspired Models and Technologies for Ambient Information Society (BioAmBIS 2010 workshop), Boston, MA, USA, December, 2010.
2. K. Habibul Kabir, Masahiro Sasabe, and Tetsuya Takine, “Evolutionary Game Theoretic Scheme for Stable and Resilient Data aggregation in DTNs,” International Journal of Autonomous and Adaptive Communications Systems (IJAACS), 2013.
3. 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.3, pp.490–500.
4. 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
5. H. Ohtsuki and M. A. Nowak, “The replicator equation on graphs. *J. Theor. Biol.*, 243(1):86–97, 2006.”
6. Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., Weiss, (2003) ‘Delay tolerant network architecture’, *IETF RFC 4838 draft*.
7. Fall, K. (2003) ‘A delay-tolerant network architecture for challenged Internets’. *Proceedings of ACM SIGCOMM 2003*, pp.27–34.
8. K. Fall and W. Hong, “Custody transfer for reliable delivery in delay tolerant networks.” Tech. report, in Delay Tolerant Networks, Intel Research, BerkeleyTR-03-030, 2003.
9. Guo, H., Li, J., Washington, A. N., Liu, C., Alfred, M., Goel, R. et al, “Performance Analysis of Homing Pigeon based Delay Tolerant Networks In: Military Communications Conference”, 2007. MILCOM 2007. IEEE, pp. 1-7 (2007)
10. Kevin, F, “A delay-tolerant network architecture for challenged internets,” In: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications Vol. 33, pp. 27-34. ACM (2003)
11. Warthman, F, “Delay Tolerant Networks (DTNs) -A tutorial”. Report No. 1.1, (2003).

12. Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K. et al, “Delay-Tolerant Networking Architecture.” <http://tools.ietf.org/html/rfc4838> (April 2007) [cited in 2008-06-05].
13. Scott, K.,Burleigh, S, “Bundle protocol specification”, RFC 5050, <http://www.rfc-editor.org/rfc/rfc5050.txt> (November 2007) [cited in 2008-06-05].
14. Ariyakhajorn, J., Wannawilai, P.,Sathitwiriyawong, C, “A Comparative Study of Random Waypoint and Gauss-Markov Mobility Models in the Performance Evaluation of MANET In: Communications and Information Technologies”, 2006. ISCIT '06. International Symposium on, pp. 894-899 (2006)
15. Seth, A., Kroeker, D., Zaharia, M., Guo, S.,Keshav, S, “Low-cost communication for rural internet kiosks using mechanical backhaul In: Proceedings of the Intermittent connection effect in the Message Ferry Delay Tolerant Network.”
16. 12th annual international conference on Mobile computing and networking, pp. 334-345. ACM (2006)
17. Demmer, M., Brewer, E., Fall, K., Sushant Jain, Melissa Ho, Patra, R, “Implementing Delay Tolerant Networking.” (Intel Research Berkeley Technical Report December 2004).
18. Xiaoyan, H., TaekJin, K., Mario, G., Daniel Lihui, G.,Guangyu, P, “A Mobility Framework for Ad Hoc Wireless Networks In: Proceedings of the Second International Conference on Mobile Data Management”, pp. 185-196. Springer-Verlag (2001)
19. Bai, F., Sadagopan, N.,Helmy, “A IMPORTANT: a framework to systematically analyze the Impact of Mobility on Performance of Routing Protocols for Adhoc Networks.” In: INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE Vol. 2, pp. 825-835 (2003)
20. Jing, T., Joerg, H., Christian, B., Illya, S.,Kurt, R. “Graph-Based Mobility Model for Mobile Ad Hoc Network Simulation In: Proceedings of the 35th Annual Simulation Symposium”, pp. 337-344. IEEE Computer Society (2002)
21. J. Kurose and K. Ross, “Computer networking. Pearson Addison Wesley”, 2005.
22. A. S. Tanenbaum, Computer Networks 4th Edition. Prentice Hall, 2003.
23. F. Warthman, “Delay-tolerant networks (dtns): A tutorial,” 2003.
24. J. Kangasharju, K.W. Ross, and J.W. Roberts, “Performance evaluation of redirection schemes in content distribution networks”. Computer Communications, 24(2):207–214, 2001.

25. A. Shaikh, R. Tewari, and M. Agrawal. “On the effectiveness of DNS-based server selection”. In Proc. of IEEE INFOCOM 2001, Anchorage, AK, USA 2001.
26. S. Gadde, J. Chase, and M. Rabinovich. “Web caching and content distribution: A view from the interior”. In Proc. of the 5th International Web Caching and Content Delivery Workshop, May 2000.
27. M. Koletsou and G. M. Voelker. “The Medusa proxy: A tool for exploring user-perceived web performance”. In Proc. Of the Sixth Int. Workshop on Web Caching and Content Distribution, June 2001.
28. B. Krishnamurthy, C. Wills, and Y. Zhang “On the use and performance of content distribution networks”. In Proc. Of SIGCOMM IMW 2001, Nov. 2001.
29. K. L. Johnson, J. F. Carr, M. S. Day, and M. Frans Kaashoek. “The measured performance of content distribution networks”. Computer Communications, 24(2), 2001.
30. Perkins, Charles E, “Ad hoc networking”, Addison-Wesley Professional, perkins2008ad, 2008.
31. Kopp, Carlo, “Ad Hoc Networking”, Systems Journal, kopp1999ad, pp: 33—40, 1999.
32. K. Fall, S. Farrell, “DTN: An architectural retrospective”, IEEE Journal on selected Areas in Common, Vol.26, no.5. pp. 828-826, June 2008.
33. Cerf, Vinton and Burleigh, Scott and Hooke, Adrian and Torgerson, Leigh and Durst, Robert and Scott, Keith and Fall, Kevin and Weiss, Howard, “Delay-tolerant networking architecture”, cerf2007delay, RFC4838, April, 2007.
34. Lloyd Wood, et.al, “Use of Delay Tolerant Networking Bundle Protocol from Space”, IAC – 08 – B2.3.10, Global Government Solutions Group, Cisco Systems, UK.
35. M. Seligman, K. Fall, and P. Mundur, “Alternative custodians for congestion control in delay tolerant networks,” inProc. ACM SIGCOMM Workshop on Challenged Networks. New York, NY, USA: ACM Press, 2006, pp. 229–236.
36. Hervé Ntareme, Marco Zennaro, Björn Pehrson, “Delay Tolerant Network on smartphones: Applications for Communication challenged areas”, published in Extremecom 2011, Brazil, September 2011.
37. R. Handorean et al., “Accommodating Transient Connectivity in Ad Hoc and Mobile Settings,” Pervasive 2004, Apr. 21–23, 2004, Vienna, Austria, pp. 305–22

38. W. Zhao et al., “A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks,” Proc. 5th ACM Int’l. Symp. Mobile Ad Hoc Net. And Comp., ACM Press, 2004, pp. 187–98.
39. Campos, C.A.V., Otero, D.C. & de Moraes (2004, March), Realistic individual mobility Markovian models for mobile ad hoc networks," Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE, vol.4, no., pp.1980-1985 Vol.4, 21-25.
40. Zhao, W., Ammar, M., Zegura, E, “A message ferrying approach for data delivery in sparse mobile ad hoc networks”. In. ACM Mobi Hoc, pp.187–198 (2004).
41. M. Papadopouli and H. Schulzrinne, “Performance of data dissemination among mobile devices,” Encours de soumission, 2001
42. A. Doria, M. Uden, and D. Pandey, “Providing connectivity to the saami nomadic community,” generations, vol. 1, no. 2, p. 3, 2002
43. J. Hofbauer and K. Sigmund, “Evolutionary Games and Population Dynamics”. Cambridge University Press, 1998.
44. E. Lieberman, C. Hauert, and M. Nowak, “Evolutionary dynamics on graphs,”Nature, vol. 433, no. 7023, pp. 312–316, 2005.
45. M. A. Nowak, Evolutionary dynamics: exploring the equations of life. Harvard University Press, 2006.
46. H. Ohtsuki and M. A. Nowak, “The replicator equation on graphs,” Journal of Theoretical Biology, vol. 243, no. 1, pp. 86–97, 2006.
47. H. Ohtsuki and M. Nowak, “Evolutionary stability on graphs,” Journal of Theoretical Biology, vol. 251, no. 4, pp. 698–707, April 2008.
48. 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.
49. L. Samuelson, “Evolutionary games and equilibrium selection” . MIT Press, 1997.
50. K. Habibul Kabir, Masahiro Sasabe, and Tetsuya 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. Cited: 4.
51. 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 87 Algebra, Control and Optimization of American Institute of Mathematical Science (AIMS), vol. 1, no. 4, pp. 563-576, December, 2011.

52. K. Habibul Kabir, Masahiro Sasabe, and Tetsuya Takine, “Evolutionary Game Theoretic Scheme for Stable and Resilient Data aggregation in DTNs,” International Journal of Autonomous and Adaptive Communications Systems (IJAACS), 2013 (17 pages). Cited: 7.