Spring 2018

Reputation and credit based incentive mechanism for data-centric message delivery in delay tolerant networks

Himanshu Jethawa

REPUTATION AND CREDIT BASED INCENTIVE MECHANISM FOR DATA-CENTRIC MESSAGE DELIVERY IN DELAY TOLERANT NETWORKS

by

HIMANSHU AVINASH JETHAWA

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN COMPUTER SCIENCE

2018

Approved by

Dr. Sanjay Madria, Advisor
Dr. Zhaozheng Yin
Dr. Haoyi Xiong

PUBLICATION THESIS OPTION

This thesis consists of the following articles that has been published/communicated as follows:

Paper I: Pages 12-43 have been communicated to 19th IEEE International Conference on Mobile Data Management (MDM), 2018.

Paper II: Pages 44-55 been published on pages 2551-2554 in 37th IEEE International Conference Distributed Computing Systems (ICDCS), 2017.

ABSTRACT

In a Data-centric Delay Tolerant Networks(DTNs), it is essential for nodes to cooperate in message forwarding in order to enable successful delivery of a message in an opportunistic fashion with nodes having their social interests defined. In the data-centric dissemination protocol proposed here, a source annotates messages(images) with keywords, and then intermediate nodes are presented with an option of adding keyword-based annotations in order to create higher content strength messages on path toward the destination. Hence, contents like images get enriched as there is situation evolution or learned by these intermediate nodes, such as in a battlefield, or in a disaster situation. Nodes might turn selfish and not participate in relaying messages due to relative scarcity of battery and storage capacity in mobile devices. Therefore, in addition to content enrichment, an incentive mechanism is proposed in this thesis which considers factors like message quality, battery usage, level of interests, etc for the calculation of incentives. Moreover, with the goal of preventing the nodes from turning malicious by adding inappropriate message tags in the quest of acquiring more incentive, a distributed reputation model (DRM) is developed and consolidated with the proposed incentive scheme. DRM takes into account inputs from multiple users like ratings for the relevance of annotations in the message, message quality, etc. The proposed scheme safeguards the network from congestion due to uncooperative or selfish nodes in the system. The performance evaluation shows that our approach delivers more high priority and high quality messages while reducing traffic at a slightly lower message delivery ratio compared to ChitChat.

ACKNOWLEDGMENTS

I would like to convey gratefulness to all those who helped me on this thesis. I would like to thank my advisor Dr. Sanjay Kumar Madria, who gave me a wonderful opportunity to work on this research. His suggestions, feedback and encouragement carried me through strenuous times. His intellect and expertise in this field of research is unparalleled. Secondly, I would also like to express my gratitude to Dr. Zhaozheng Yin and Dr. Haoyi Xiong for serving on my thesis committee, and for taking time to review this work.

I am grateful to all my lab mates and friends for their professional help and inputs. This project is partially funded by the Air Force Research Laboratory (AFRL) in Rome, New York, and I express my gratitude to them as well.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

The ever-rising use of mobile devices in the world has made it possible to form a highly effective and efficient mobile p2p network. The concept of delay tolerant networks was introduced because of the lack of infrastructure in disaster affected areas and extra-terrestrial networks. Due to the realization that it is not possible to deliver messages instantaneously in such scenarios, algorithms that allow the incorporation of the delay were developed. NASA and other organizations were funded to develop a proposal for Interplanetary Internet(IPN). The initial architecture was proposed keeping in mind the significant delays and packet corruption in deep space communications. Over time, some of the ideas proposed for IPN were adapted to and the term “delay-tolerant networking” was coined.

The causes for disruption can be limits of wireless radio range, sparsity of mobile nodes, energy resources, attack, and noise. To counteract these challenges, routing protocols were proposed. One of the first few protocols defined was epidemic dissemination of data. In this protocol, every device transmits all the data possessed to every other encountered device and vice-versa. Although this algorithm achieves the highest delivery ratio, the overhead it imposes is huge. This overhead is measured in terms of energy consumption and used data space at the node level and overall traffic at the network level. Other routing mechanisms were therefore proposed and implemented in different scenarios by taking into account the suitability of trade-off between throughput and overhead.

As time evolved, researchers realized that the problems faced in disruption tolerant networks appear in many other scenarios. Therefore, issues involved in vehicular ad-hoc networks, rural communication, conferences, etc. can also be solved by the DTN architecture.

Figure 1.1 below shows the architecture of DTN. Nodes can be stationary or mobile. A DTN node shares the data with another node when a successful connection is established. When an intermittent link goes down, it another path can be used to deliver a message. Multiple paths can therefore be used to share the message. This is why it is also called as opportunistic communication.

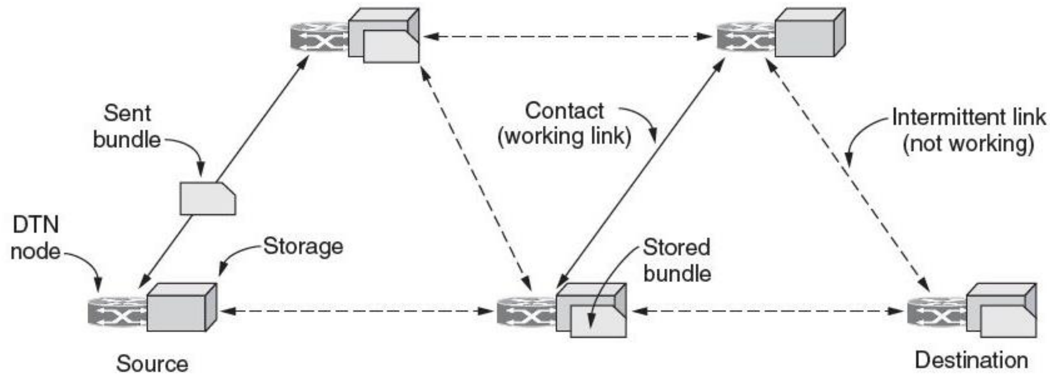

Figure 1.1. DTN Architecture

The following subsection defines routing.

1.1. ROUTING IN DTN

As the traditional routing algorithms like Open shortest path first(OSPF), Gateway Routing Protocol(GTP) cannot be used directly in DTNs because they require a continuous end-to-end connectivity, other schemes have been proposed. Broadly, the algorithms can be classified as node-centric and data-centric. Node-centric algorithms are further classified as flooding-based and forwarding-based.

A few examples of flooding-based are Epidemic, Direct-contact, Two-hop-relay, and Spray-and-Wait. In Direct-Contact routing algorithm, the source node directly forwards a bundle to the destination node. Some variants of epidemic routing are priority-based and immunity-based. In two-hop relay, a message will be delivered to destination if source and destination are within two-hops reachability. In Spray-and-Wait, replicas of a message are distributed to an optimal number of nodes rather than all the nodes. This can also be considered another variant of Epidemic routing.

Some of the forwarding-based algorithms are NECTAR, Source Routing, and Per-Hop routing. NECTAR maintains a neighborhood index table at each node which stores the information about the meeting frequency of encountering nodes in the network. Nodes with higher index are forwarded bundles. Source Routing has two phases, viz., route discovery and route maintenance. Route discovery phase discovers the route from source to destination with intermediate nodes appending its address in the discovery packet. Route maintenance phase throws a route error when a link is broken. In per-hop routing, decisions about forwarding to the next hop are made individually by each node. The next subsection defines data-centric routing.

1.2. DATA-CENTRIC ROUTING

Given a vast amount of information to disseminate, it is important to prioritize the data delivery to the data which is relevant to the nodes. Therefore, it makes more sense for the nodes to analyze the content of the received bundles before making decisions about the next hop to forward rather than just blindly forwarding the data without knowing the data itself. A security issue of what if a relay uses the received data for malicious purposes arises. In such cases, data can be encrypted but a set of metadata keywords can be defined to tag the data with.

Data-centric routing was proposed to further reduce the overhead in the network while also giving a reasonable throughput with only the relevant contextual data being forwarded and ultimately delivered. Examples of Data-centric routing algorithms are CEDO[1] and ChitChat[2].

CEDO is acronym for Content Centric Dissemination algorithm. It solves the problem of how nodes should replicate content so that the network throughput is maximized. The idea is to allow nodes to make requests for content at random times. The request is tagged with TTL, which when expires, the request is deleted from the entire network. If a node m comes in contact with node n and node n has content in its buffer that m requested, m can retrieve the content from node n.

ChitChat, on the other hand, is a bit intricate and sophisticated compared to CEDO. Every node has predefined interests that act as subscription keywords. Keywords are mapped with the corresponding strength which rises or falls depending upon the encountering nodes' keyword strengths. When a node m encounters node n, node m decides to forward the content possessed by it to node n if the keyword strength in n for the content equals or exceeds that in node m.

1.3. MOTIVATION

Having the most efficient and effective routing mechanism in place does not mean that the nodes will start cooperating with each other in the process of relaying. This is owing to the fact that the mobile nodes have limited battery power and also relatively small storage space for storing the in-transit messages.

The main problem we solve in this paper are of identification of selfish nodes and eliminating the traffic due to them. We introduce a concept of content-enrichment as well which allows the nodes to make the in-transit packets richer in metadata information. Apart from this, there is a need for a reputation based mechanism. All these three problems and their novel solutions are described briefly in the subsections below:

1.3.1. Need to Identify Selfish Nodes. Selfish nodes are defined as the nodes which either refuse from receiving incoming packets for relaying, dropping existing packets even before the time-to-live(TTL) is not over and the message has not been forwarded, and denial of participation in relaying the acquired messages. An important characteristic of such nodes is that they receive the messages intended for them and not receive the messages for relaying. This might be accomplished by switching on the communication medium when in need and switch it off when not. The existence of these category of nodes is a very serious problem in delay tolerant networks because these networks rely on nodes forwarding the data to the encountered hops. If the message bundles are not forwarded to the encountered nodes, this might result in unsuccessful delivery of those bundles.

We have therefore defined an incentive mechanism which motivates all nodes to participate in the message transactions. This mechanism ensures fairness to all the devices. The developed mechanism is primarily based on credit-based. Every nodes is assigned a pre-defined number of tokens initially. All nodes are assigned the same number of tokens. They can then use these tokens to pay as compensation for message transactions.

Consider a scenario of a selfish node which receives the messages it wants and does not participate in forwarding. In this case, the selfish node will end up paying all of its assigned incentive tokens and be left with zero tokens eventually. Unless the node participates in relaying and gains more tokens to pay for the content it requires, the node will not be able to receive the interesting content. Therefore, the traffic due to those selfish nodes will be curbed.

1.3.2. Content Enrichment. Consider a scenario where nodes can take a peek into the data being relayed by them and happen to have supplementary information about the content. If the metadata with which this bundle is tagged does not reflect the above additional information about the content of the bundle, it makes sense for these relay nodes to add more metadata keywords. There is, therefore, an elevated possibility for the number of destinations to be risen. Additionally, the added information can help the nodes to have a much better situational awareness. This adding of metadata keywords is what we call content enrichment.

The question which needs to be addressed is why a user might be feeling generous enough to enrich the content. In other words, *what is the profit of a user who is adding more metadata keywords?* A rational user adding the keywords cannot be certain that other users will do the same. Therefore, even the generous users might lose the motivation to do so. In order to facilitate the content enrichment, we define in the scheme that the user adding relevant metadata keywords can attain more incentive tokens from the destination than what were promised to it by the sender. This scheme ensures fairness and validates that the users are paid proportionately to their contribution.

1.3.3. Distributed Reputation Model. Malicious nodes are defined as the ones which either generate poor quality messages or add irrelevant keywords to the passing bundle. The motivation behind this behavior can be to gain higher incentive tokens. For example, consider a node which acquired a message consisting of an image of a tree tagged with the keyword “tree”. Based on our content enrichment scheme, this node can add more irrelevant keywords. Let us say the node adds keywords “car”, “books” and “building”. Let us say this malicious node delivers the message to the nodes having the subscription keywords “car” and “building”. Since the message transaction is automatic, the destinations for the delivered messages provide extra incentive to the malicious nodes. The mobile node is limited with computational power and memory, it definitely cannot execute the machine learning algorithms on its own. Therefore, user intervention is required to identify the malicious nodes and notifying the other encountered nodes about the identified malicious nodes.

A user can make decisions about quality of message and relevance of the additional keyword annotations post-reception. Therefore, a rating can be added for individual nodes and then shared with other nodes in order to spread the reputation of the nodes network wide and enabling other nodes to avoid receiving from malicious nodes. Furthermore, the decisions of incentive awarding are done by taking into consideration of the reputation of nodes. In our approach, a percentage of incentive is provided to deliverers with reputation value below a certain threshold.

2. RELATED WORK

There has been sizable amount of work on Incentive mechanism in Delay Tolerant Networks. Since the incentive mechanisms that we have developed is based on credit and reputation, it is important to note down those kinds of previous and related works. Thus, the following subsections manifest these two types of work in detail:

2.1. CREDIT BASED INCENTIVE MECHANISM

These kind of incentive mechanisms are based on rewarding nodes for generating or relaying messages. There are two broad classification in these: i) where source pays the incentive tokens, and ii) where destination pays incentive tokens for the relayed message.

Mobicent[3],PI[4],[5], MuRIS[6], TFT[7] are some of the works that are related to varying degrees with credit-based work in the proposed scheme in this thesis.

In MobiCent, client's payments and the relays' rewards are set with a goal that nodes will behave truthfully. Therefore, packets will always be forwarded by nodes without adding phantom links, and the nodes will not let the contact opportunity go to waste unless the reward is not adequate or the decisions of the underlying mechanism dictate that. The underlying routing protocol will find the best available path for a message delivery. There are three kinds of nodes in the Mobicent architecture: i) Trusted third party which stores key information for all nodes and provides verification and payment services. ii) Helpers, which are static or mobile nodes, relay content via short range, high speed and intermittent communication links. iii) Mobile clients that are the destinations which have high-bandwidth intermittent links for data transfer and highly available but low bit rate links for control messages.

In PI, a source attaches an attractive as well as fair incentive to a bundle. In this system, there is a trusted authority(TA) which does not participate in bundle forwarding but rather it performs clearance of credit and reputation for DTN nodes. Every DTN node before joining the network registers itself with the TA to get its personal credit account(PCA) and a personal reputation account(PRA).When a node later establishes a connection with the TA, it can make requests to TA for credit and reputation clearance. An intermediate node when participates in bundle forwarding, it can get credits from the source. At the same time, it can get reputation from the TA. Therefore, PI relies on a centralized trusted authority for reputation management.

In [5], the authors have defined a two-hop scheme in which a source generates a message, forwards it to a relay along with a promise of incentive and a relay when delivering a message receives the promised incentive from the destination. They have three settings that they evaluate based on two parameters: i) Number of circulated copies for a bundle and ii) Duration of time for which those copies have been present in the network. The three settings are: i) full information, ii) partial information, and iii) no information. A source when forwarding a bundle to a relay along with a promise of the incentive, it gives the relay either full, partial or no information. The full information implies that the source tells the relay the number of copies that it has distributed before and also the time for which those copies have been in the network. The partial information signifies that the source only gives out the information about the number of copies. The no information setting means that the source does not give any information about the copies in circulation. A relay therefore can decide if the incentive being promised to it is fair or not in terms of whether it will be able to deliver the message first. The reason behind this is that in this scheme, a relay to forward a message to a destination only receives the promised incentive from the destination if it is a first deliverer to that destination. The relay however can deliver the copies of the messages to multiple destinations and receive the incentive promise.

In [6], MuRIS, an incentive driven information sharing in DTNs is proposed. It dynamically constructs efficient multicast delivery paths for multiple destinations interested in the same data item. An incentive mechanism motivates the uncooperative nodes so that they collect rewards associated with their forwarding efforts. There are two phases in MuRIS, viz., Information collection stage and Data Forwarding stage. In the warmup stage of Information collection, nodes use probe/receipt messages to learn about paths from publishers to various subscribers in the network. In addition to this, when two nodes encounter, they exchange the path information that is gathered by them and subsequently update their paths. Based on this, each nodes constructs a feasible path set. A closeness vector is also constructed based on encounter histories. In the Data forwarding stage, nodes exchange the data items of interests to each other. For the data items that could be forwarded by the other node, a node quantifies the reward for forwarding based on the feasible path set and the closeness vector. A data item is forwarded to the other node only if the path via the other node has a promise of providing the highest expected reward.

In [7], the authors have proposed a Tit-For-Tat scheme which is an incentive-aware routing in DTNs. The main goal of TFT is to maximize the delivered traffic within a certain time frame. Their routing protocol consists of the following three components: (i) Link State Dissemination module in which every node periodically exchanges link state, (ii) Route computation module in which each source computes the forwarding paths based on link state and uses source routing to send its traffic, (iii) Acknowledgement dissemination module upon receiving data, each destination sends ACK via flooding and the source uses it to update its TFT constraints for the next interval. It is built on source routing because source routing can be used to optimize customized parameters for a source. The main assumption in this paper is that the link state is disseminated faithfully. They focus to make the data-plane incentive compatible. Acknowledgements can provide useful feedback required by TFT. Specifically, every node receiving the ACK first verifies the integrity of the attached source route and then checks if its identifier is present in the relay list. If it is, then the node increments its local TFT counters to indicate that the next node in the list successfully relayed a packet for it. Credit is only given to relay nodes on the forwarding path.

2.2. REPUTATION BASED RELAY COORDINATION

Reputation based coordination schemes are the ones in which nodes make use of the reputation values of the encountered nodes to determine whether to transfer a bundle to those nodes. In distributed reputation models, nodes take decisions by themselves after gathering information from previously encountered nodes, without intervention or assistance from a centralized entity. There are various ways to determine malicious nodes. The rule of thumb is the lower the reputation of a node, the more malicious that node is. Some of the related approaches of distributed reputation metric in delay tolerant networks are given below.

REPSYS[8] is a recently proposed robust and distributed system for delay-tolerant networks. It utilizes a modified Bayesian approach to identify malicious nodes. A reputation rating R_{ij} between two nodes i and j is managed by the reputation module. It is updated when either the first-hand information is updated or the second-hand information is found to be valid. REPSYS is robust against false ratings and it is also efficient at detecting nodes' misbehavior. REPSYS is robust because despite taking into account all the available information, it is resilient against false accusations and praise. It is also distributed because the decision to interact with another node is made entirely by each node. Bayesian decision theory is used to classify the nodes after taking into consideration all the available information. It is based on a Bayesian approach that uses the Beta distribution, and can be integrated with any DTN routing protocol. There are three modules in REPSYS: reputation module (reputation collection module, reputation evaluation module), trust module and routing decision module (that uses Bayesian classification).

In [9], the authors have proposed a trust-based framework for data forwarding in opportunistic networks. They incorporate idea of a watchdog component at a node which is responsible for monitoring the behavior of other nodes. Whenever a node, say X, forwards a message to node Y which in turn forwards the message to node Z, node X looks out for Positive feedback messages(PFM) from the destinations(Z in this case). The watchdog component maintains two counters to determine the forwarding behavior of other nodes. Each node (e.g., node X) which sends out data to its next-hop forwarders, will keep recording for each of those forwarders how many PFMs corresponding to data sent out have and have not come back by using two counters, respectively. A next-hop forwarder is suspicious if no PFMs are received for that forwarder. Therefore, there is no evidence of good forwarding behavior of that node. The no reception of PFMs due to the intermittent connectivity nature of opportunistic networks is also taken into account with Beta probability distribution function.

In [10], an iterative algorithm for trust management and adversary detection is proposed. Service Providers(SP) are defined as nodes that participate in data forwarding whereas raters(R) are the ones that rate the data forwarders. The idea is based on Low Density Parity Code Check (LDPC). A bipartite graph is drawn consisting of SPs and Rs as vertices. If R has a rating for an SP, an edge is drawn between the two. The algorithm

(ITRM) is executed at fixed time intervals. In the first iteration of the algorithm execution, the weight of the edge between an R and an SP is the rating that R has for that SP. A fading parameter w is defined. In the iterations henceforth, when a new rating between ith rater and jth SP arrives, the weight of the edge is recalculated as an average of the new rating and the old rating multiplied with the fading parameter. The decision of whether a node is good or bad is taken by a Beta probability distribution function. The problem of whitewashing, i.e., nodes with low ratings cancelling their account and then signing in again with a new ID in order to get a good rating, is also handled.

Another cooperative watchdog system(CWS) to detect and avoid selfish nodes in the network in Vehicular Delay Tolerant Networks is proposed for in [11]. In this system, a node assigns a reputation value to each other node. Each time nodes get a contact opportunity, the CWS updated the reputation score via three modules, viz., classification, neighbors' evaluation and decision modules. Nodes are classified into different types based on their reputation scores and the classification module calculates each node's cooperative value. The cooperative value is then used by the decision module to determine whether to punish or reward the encountered node. Neighbors' evaluation module determines how neighbors evaluate a node's reputation on the network. Neighbors opinions are inquired and based on that, the reputation values are assigned. At the end of a contact opportunity, the decision module updates the rating for the encountered node based on the input from the other two modules.

PAPER

I. REPUTATION AND CREDIT BASED INCENTIVE MECHANISM FOR DATA-CENTRIC MESSAGE DELIVERY IN DELAY TOLERANT NETWORKS

Himanshu Jethawa and Sanjay Madria
Department of Computer Science
Missouri University Of Science and Technology, MO 65401
Email: {hj5y3@mst.edu, madrias@mst.edu}

ABSTRACT

In Delay Tolerant Networks (DTNs), to ensure successful message delivery, contribution of mobile nodes in relaying in an opportunistic fashion using social interests is essential. In our proposed data-centric dissemination protocol here, messages (images) are annotated with keywords by the source, and then intermediate nodes are presented with an option of adding keyword-based annotations to create higher content strength messages en-route toward the destination. Therefore, contents like images get enriched as the situation evolves or learned by these intermediate nodes, such as in a disaster situation, or in a battlefield. Due to limited battery and storage capacity in mobile devices, nodes might turn selfish and not participate in relaying messages. Thus, additionally, an incentive mechanism is proposed in this paper which considers factors like message quality, level of interests, battery usage, etc for the calculation of incentives. Moreover, in order to prevent the nodes from turning malicious by adding inappropriate message tags in pursuit of acquiring more incentive, a distributed reputation model (DRM) is developed and integrated with the proposed incentive scheme. DRM takes into account inputs from multiple users like ratings for the message quality, relevance of annotations in the message, etc. The proposed scheme thus ensures avoidance of congestion due to uncooperative or

selfish nodes in the system. The performance evaluation shows that our approach delivers more high priority and quality messages with reduced traffic with a slightly lower message delivery ratio compared to ChitChat, where a source forwards a message to intermediate nodes, which meet or exceed the matching strength of keyword-based interests.

1. INTRODUCTION

Delay tolerant networking (DTN) is a networking paradigm characterized by frequent disconnections between nodes in the network hence resulting into lack of end-to-end connectivity[1]. Message delivery in this scenario is achieved by utilizing the resources of multiple nodes which act as relays in the network. These nodes relay content in order for the final destination to receive it. DTN was initially proposed for inter-planetary networks and disaster relief team networks. Recently, it has also been applied to environments such as social networks and vehicular networks, and more recently in defense applications. A key challenge in DTNs is achievement of high message delivery ratios with finite lifetimes [2][3][4].

In traditional DTNs, nodes are assumed to help other nodes for packet forwarding. But in a real scenario of DTN deployment, the relay nodes could abstain from cooperation due to a limited storage capacity and battery lifetime. As mobile nodes are managed by autonomous parties, the assumption that all the nodes are cooperative is unrealistic, and at the same time, in defense applications, though the cooperation is not an issue, but controlling the congestion is important. Therefore, incentive schemes [7] have been proposed to foster cooperation among participants, and control the congestion in DTN, which help in relaying messages.

One of the important aspects ignored by DTN routing associated with incentive schemes is that messages are taken and delivered as black box by relays, but do not use the knowledge of the relaying users in making the content richer by adding semantics in terms of annotations. Consider an application like military, where in-transit users (soldiers) may provide much more information about say images which are in transit through DTNs. This can provide much better situational awareness in other applications also such as disaster response and recovery where situation evolves as time progresses.

Another issue usually ignored in DTNs is the absence of feedback messages, thereby incurring large delays. In fact, the exchange of rewards between relays should not require feedback messages. In order to overcome the lack of feed-backs, the proposed mechanism assumes that a relay receives a positive reward if and only if it is the first one to deliver the message to the corresponding destination. In a similar fashion, [6] proposed a two hop incentive scheme in which the source promises incentives to each and every relay but informs them that only the first one who deliver the message to the destination, receives the incentive.

In this paper, we present a novel incentive mechanism taking into consideration reputation of nodes as well as addition of tags in enabling content-centric sharing of data such as images in DTNs. The parameters considered for incentive calculations are categorized as software and hardware factors. Software parameters are message size, message quality, interest levels of connected devices in the message, rank of the users possessing the connected mobile devices and priority of messages. Hard- ware factor taken into consideration is the energy consumption of a device delivering the message to the destination.

One key challenge in developing any incentive mechanism is the presence of malicious nodes which try to game the system by generating or relaying seemingly relevant message to the destination at an abstract level while also helping themselves in attainment of higher incentives. For example, consider a message consisting of an image of only a ”garden”. A source might annotate this message with a keyword “parking lot” but there is no parking lot in the image. In this case, since the incentive mechanism is developed based on the message annotations, a destination with an interest represented by ”parking lot” pays for false annotations on reception of the message. This problem is tackled by introducing a distributed reputation metric (DRM).

Most of the existing works on distributed reputation metric depends on the existence of a centralized entity for reputation management. Even if it is distributed like recent work in [25], only the behavior of the nodes in terms of routing is evaluated. It is certainly not feasible for the scenarios involving DTNs which we are considering where nodes may modify the contents maliciously. We also took into consideration different factors for the DTN scenario and developed a novel distributed reputation metric (DRM). The proposed DRM requires human judgement and input on each message content and each device owner can give a rating for message content quality and truthfulness of message annotations.

To the best of our knowledge, no incentive mechanism is designed for data dissemination in DTN which considers content enrichment of the message (images) as others are routing-centric. Our proposed scheme works on top of our state of the art routing called ChitChat routing [5] for data-centric message delivery in DTNs. We selected ChitChat as it achieves better message delivery ratios with other competitive DTN routing algorithms, by introducing the concept of transient social relationships (TSRs). Thus, our current scheme further improves data dissemination throughput within ChitChat integrated with the content-based incentives and reputation schemes. Note that our proposed scheme can be integrated with any other DTN routing scheme.

In summary, the following contributions are made here:

• We propose content-based data dissemination by proposing the modification to the routing phase of the ChitChat algorithm by incorporating incentives associated with the relaying as well as for making in-transit contents enriched by asking intermediate nodes to add meaningful annotations (to earn incentives) to improve the throughput and data quality with respect to the content delivery.

• Incorporating distributed reputation mechanism of users in DTNs, where incentives earned are linked with reputations of users as well as quality of data.

•The performance evaluation of our credit and reputation based scheme is performed using ONE [23] simulator.

2. RELATED WORK AND BACKGROUND

2.1. RELATED WORK

In literature on DTNs [8][9] , several incentive schemes have been recently proposed. For example, Mobicent[11] is a credit-based incentive system in which credit and cryptographic techniques are integrated to counteract the edge insertion and edge hiding attacks among nodes. SMART[13] is a secure multilayer credit-based incentive scheme for DTNs. In SMART, layered coins are provided as incentives to selfish DTN nodes for bundle forwarding. In [10], Tit-for-Tat (TFT) is used to design an incentive-aware routing protocol in which selfish DTN nodes are allowed to maximize their individual utilities in conformity with TFT constraints. In [18], authors proposed an incentive driven dissemination scheme in which nodes are encouraged to cooperate while choosing delivery paths that can reach maximum number of nodes possible with fewest transmissions. PI [12] attaches an incentive on the sending bundle to stimulate the selfish nodes to cooperate in message delivery. MobiGame [14] is an incentive scheme for DTNs based on user centricity and socially aware reputation. [15] proposes socially selfish routing in DTNs, where a node considers social willingness to determine whether or not packets should be relayed to the nodes in the vicinity. Authors in [16] formulate nodal communication as a two-person cooperative game for a credit-based incentive scheme to promote nodal collaboration. RELICS [17] is a energy-aware cooperation based incentive mechanism for selfish DTNs, in which a rank metric is designed to quantify the transit behavior of a node. [19] is a credit-based incentive system using the theory of Minority Games [20] in order to attain coordination in distributed fashion. This mechanism considers the realistic case when the cost for taking part in the forwarding process varies with the devices technology or the users habits.

In [21], detection of faulty sensors in a DTN node of a wireless sensor network is performed by using outlier detection algorithm. DISARM [22], another distributed reputation model, takes into account social relations amongst agents in a multi-agent environment. In DISARM, the agents draw reasonable conclusions from incomplete and possibly conflicting information based on factors like correctness, transaction value and so on. A Robust and Distributed Reputation System for Delay-Tolerant Networks [25] has been proposed which considers interaction with nodes, feedback messages and false ratings. [26] uses trust-based framework to more accurately evaluate an encounter's delivery competency. A graph based iterative algorithm motivated by the prior success of message passing techniques for decoding low-density parity-check codes over bipartite graphs is used for adversary detection in [27]. A cooperative watchdog system is proposed in [28] in which nodes exchange reputation of previously encountered nodes in the network in order to detect misbehaved nodes.

2.2. BACKGROUND

This section provides overview of the underlying routing algorithm, ChitChat, used in our work.

In ChitChat, nodes are users with small pocket devices that are equipped with ChitChat system, which automatically connect to other devices that are within the communication range. Each user has their own social profile, i.e., a group of interests specified by semantic keywords. Messages are also annotated with appropriate metadata keywords. When two nodes connect with each other, they chitchat to exchange the following two kinds of information: (i) direct social interests, which is the metadata, or the set of keywords, that describe the encountered node's interests, roles, and responsibilities (e.g. social interests such as "photography" and "gourmet cooking", or role-specific metadata such as "MANET researcher", "military intelligence officer"); and (ii) transient social relationships, which is aggregated information of the social interests of the people that the node has encountered before. This approach allows the social interests to be dynamically expanded, refined and aggregated in real-time into the richer transient social relationships so as to capture multi-hop relationships. A destination for a message is defined as a device with direct interest in keywords of the message whereas a relay is defined as one with acquired interests.

ChitChat system consists of two major components with associated storage buffers: (i) Realtime Transient Social Relationship (RTSR) modeling, and (ii) Message Routing. The overall data flow in the ChitChat equipped network is as follows: When two users come within communication range, the ChitChat system first invokes the RTSR module. The RTSR module will automatically exchange the two users' current Transient Social Relationships (TSRs), resulting in an adjustment in their TSRs based on a growth-decay model. Then, the ChitChat invokes the message routing to exchange a selected subset of messages carried by the two users based on the analysis results of their revised TSRs. Following subsections define RTSR and Message routing modules in detail.

2.3. RTSR MODULE

The Real-time Transient Social Relationship modeling aims to represent the evolution of each user's social interests impacted by the people that they encounter. The flow for RTSR after two ChitChat equipped devices have established connection is: i) decay algorithm ii) exchange of decayed weights iii) growth algorithm. Initially when the interest represented by a keyword is defined for the first time by a user, it's weight is set to 0.5. Maximum allowed value for the weight is 1.

Given below are the decay and growth algorithms:

- Decay Algorithm

Given interest I for a user, the weight for I is decayed based on whether a device with shared interest I is currently connected to this user's device or not.

The variables in the following algorithm (Algorithm 1) are:

β-decay constant, Tc - current time, T_l - Latest timestamp at which a device with interest I was connected, W_n- new weight, W_p- previous weight

Algorithm 1: Decay algorithm execution in device u

procedure DECAY(u)
for all I in device u
If a device with I is connected:
$W_n = W_p$
If a device with interest I is not connected:

If I is a direct interest, then
$W_n = (W_p - 0.5)/(\beta * (T_c - T_l)) + 0.5$
Else
$W_n = (W_p)/(\beta * (T_c - T_l))$
end for
end procedure

Example execution: If device 1 is connected to device 2 and both have interest in "food coupon" with a corresponding weight of 0.6 initially and both devices had a connected device with a shared interest “food coupon" about 5 seconds ago and which is no longer connected, then new weights are calculated by the formula $W_n = (W_p - 0.5)/(\beta * (T_c - T_l)) + 0.5$, $W_n = (0.6 - 0.5)/(2 * 5) + 0.5 = 0.55$.

- Growth Algorithm

After decay phase, the updated weights are exchanged. In the following algorithm (Algorithm 2), the symbols are: ψ is an integer value out of a set of $\{1{,}6\}$ corresponding to different cases. E.g., if both u and v have I as a direct interest, value of ψ is 1. If u has a direct interest and v has a transient interest, value of ψ is 2. w_n is the newweight of an interest I in the node u whereas w_p is the old weight. $w_v(I)$ is the weight of interest I in v. T_v is the time at which v established connection. Δ is the change in weight.

Algorithm 2: Growth algorithm execution in device u

procedure GROWTH(u)
for all interests I in device u
$\Delta = 0$
for all currently connected devices v
$\Delta {+}{=} (w_v(I) * (T_c - T_v))/\psi$
end for
$w_n = min\{1, w_p + \Delta\}$
end for
end procedure

2.4. MESSAGE ROUTING MODULE

The message routing module selects better message forwarders based on their Transient Social Relationships. Moreover, each relay has a message buffer with a fixed size. After this execution is done, messages are routed according to the sum of weights of social interests in the sending device and receiving device. If the sum of weights of interests for a message in sending device is less than that in receiving device, then that message is forwarded. If $S_v > S_u$ for message M, then forward message M to device v, where u is the sending device, v is the receiving device, and S_u and S_v are the sum of weights for interests in message M in devices u and v.

3. DTN ARCHITECTURE OVERVIEW

In Figure 3.1, the overall architecture of the developed system is shown. The devices owned by Alice and Bob are deployed with the implementation of the proposed system. After establishing connection, the RTSR+DR module shares annotations and encountered devices' reputations. The message router module in Alice then selects the messages for which Bob can be destination or relay by incorporating mechanism. The same

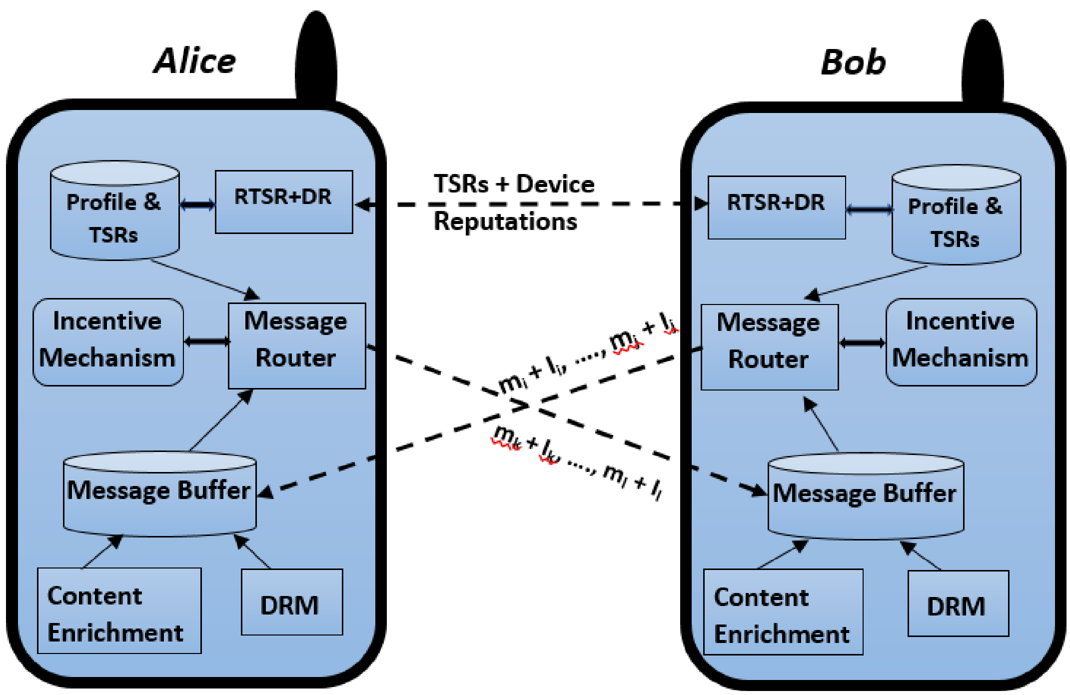

Figure 3.1. Data Flow between Two Connected Nodes

module is run by Bob. Following this, Alice and Bob have option of adding more text annotations to the received messages in message buffer by running content enrichment module. Additionally, they can run the module of proposed Distributed Reputation Model(DRM) and input ratings for the messages. The three contributions of our work are

Incentive Mechanism, Distributed Reputation Model and Content Enrichment through annotations. The three subsections included in this section discusses message format, incentive mechanism and distributed reputation model.

3.1. MESSAGE FORMAT

The multimedia message format is shown in Figure 3.2 where (i) topic implies interests (ii) location parameters such as latitude and longitude are stored as key-value

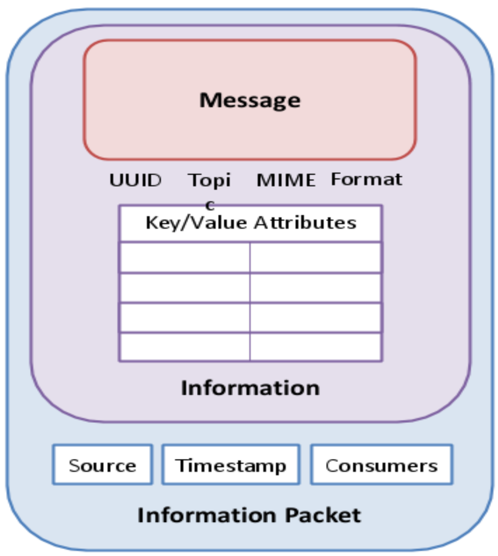

Figure 3.2. Message Format

attributes. A message in the system is an encapsulation of the multimedia data along with some metadata tags. Additionally, a UUID which is a unique identifier for the message makes sure that the message does not get duplicated in any device. A timestamp, the time at which the multimedia content for the message created is also added as a part of the message. Source and destinations, MIME and format of the message are also added.

3.2. INCENTIVE MECHANISM

The proposed incentive mechanism is credit and reputation based. This subsection explains the credit based part. All the nodes are assigned a start value of incentive tokens that can be used to perform the message dissemination as well as for making the content enriched. The factors considered for incentive determination are categorized as software and hardware. The software parameters are specific to user and content quality, whereas hardware parameters comprise of energy consumption in providing the service.

In our model, a source or a relay forwards a message to another relay node along with a promise of certain number of incentive tokens from a destination upon successful delivery of the message. In the case where there are more than one hops from a source to destination, a relay follows the exact same process that a source follows. Additionally, if a relay A transfers a message to a relay B, A receives a fraction of incentive from B if B has a very high chance of delivering the message. For example, for a message M tagged with a set of interests/keywords I{i1,i2,...in} having corresponding set of weights W{w1,w2,..,wn} to be forwarded from relay A to relay B, an average of weights A_w are computed for message for the remote node B. If A_w is greater than a certain very high pre-defined value, B offers a percentage of incentive token values to A.

Table 3.1. Symbol Descriptors

I_s	Promised incentive value due to software factors
I_h	Promised incentive value due to hardware factors
P_v	Priority level of message M to node v
R_u	Role of the sending user
I_m	Maximum incentive possible
S	Size of the message
S_m	Maximum size of messages in u
Q	Quality of the message
Q_m	Maximum quality of a message from a set of messages in u
P_s	Priority of a message set by source of the message(1-3 for high, medium, low)
w	Weight of interest for a message

Table 3.1. Symbol Descriptors(Contd.)

w_m	Maximum out of sum of weights of interests as known by u corresponding to all connected devices for a message
P_r	Reception power for a device
P_t	Transmission power for a device
L_v	Path loss
I_t	Incentive reward to a relay due to all added tags
I_{t_k}	Incentive reward to a relay due to an added tag k
R_i	Rating of the message i
R_t	Input Rating for the tags of a received message
C	Confidence of a user in the input rating of tags
C_m	Maximum possible value of confidence
R_q	Rating for quality of message
α	Weight of self-calculate rating
$r_{v,u}$	Rating of device v in device u
m_v	Message received from v
r_{m_v}	Rating of message m_v
$r_{m_v,x}$	Rating of message m_v corresponding to node x and as known by node v
r_m	Maximum device rating
I_v	Incentive awarded to node v for a message m_v

A destination can also reward an intermediate node if that node added additional relevant message annotations to in-transit messages. The motivation behind relays enriching content is, therefore, attainment of higher incentive. Finally, if a device exhausts all of its tokens, it is no longer allowed to receive messages that it itself is interested in. This results in reduced network congestion. Calculation of incentive promise is explained in following subsections, ordered as Software factors and Hardware factors.

- Software factors

The user-centric factors considered here are priority level of message defined by source, role of a user, priority level of message to the destination. The data-centric factors considered are size and quality of the message. The table above shows the parameters used in our algorithms.

Users might have different roles (R) based upon the scenario of DTN deployment. For example, in a battlefield deployment, users can be Sergeant, Soldier, etc. The user on top of hierarchy has a role 1. In this example, 1 corresponds to a user who is a Sergeant. If Soldier is the next role in the hierarchy, this category corresponds to 2, and so on. Priority level of a message M that is possessed by a user u that intends to forward it to user v(P_v) is defined as a ratio of i) sum of weights of interestsfor message M in v as known by u($\sum w$) to ii) maximum sum of weights of interests for the message amongst all the devices connected to u(w_m). P_v ensures that maximum incentive promise is provided to a node which belongs to a set of currently connected nodes and has the highest delivery probability. When two nodes connect and are in routing phase of ChitChat, a set of algorithms are executed, first of which is the following:

Algorithm 3: Calculate incentive promised from user u to user v due
to software factors
procedure calculateIncentive(u,v)
if P_v=0 and $R_u < R_v$ and $P_s = 1$(high)
$I_s = I_m$
else If $P_v <> 0$,
do $Pv = (\sum w)/(w_m)$
$I_s = (1/4 * (S/S_m + Q/Q_m) + ½ * (P_v/(R_u * P_u))) * Im$
end if
end procedure

To explain the above algorithm, consider a scenario of battlefield with user roles as ßoldier" represented by 2 and ßergeant" represented by 1. Device u belongs to a sergeant and device v to a soldier. Even if v cannot deliver a high priority message tothe destination at the current moment, it might still acquire the TSRs corresponding to the message, resulting in non-zero probability of v's ability to deliver the message. Device v is promised maximum possible incentive. Instead, if v can deliver the message, factors such as size of the message, quality of message and others are taken into account and incentive is a factor of maximum incentive possible. Greater the size of the message, greater the incentive promise to v as a long message reduces the buffer size by a higher factor compared to smaller message. To ensure that the messages generated in the system are of higher quality, high quality promises higher incentive. In the formula given above in else case, user-centric and data-centric factors each have 50 percent weight in deciding the incentive promised.

- Hardware factors

The hardware parameter taken into consideration is the power consumption in transmission of a message. If a source directly delivers a message to the destination, the incentive tokens corresponding to the amount of power consumed in transmitting that message is awarded to the source by the destination. If a relay instead delivers the message to the destination, it receives incentive tokens proportional to the amount of power consumed in receiving the message as well as forwarding of the message. I_h represents the incentive tokens' promise corresponding to hardware factor.

As battery conservation is significant issue in mobile devices, source and relay must be rewarded proportionally for the energy consumption in relaying. Friis equation [29] is used for calculating power consumption on receiver side. Incentive promise is defined as a function of power consumption. The parameters involved in Friis equation are i) transmission power, ii) receiving power, iii) distance between the connection devices, iv) Bandwidth. The receiving power can be calculated using Friis equation as follows:

$$P_r = P_t / L_v$$

where

$$L_v = (4 * \pi * R/\lambda)^2$$

R=distance between two devices and λ=bandwidth

When a source directly delivers a message to the destination, the incentive proportional to the battery consumption is only a function of transmission power and elapsed time.

$$I_h = c * P_t * t$$

where c is a constant, P_t is the transmission power of the source and t is time elapsed in delivering the message.

When a relay delivers a message to the destination, the data dissemination consumes battery in the relay in the reception of the message from another relay or source as well as in the transmission of this message to the destination. So the relay should becompensated for both of these actions. In this case, incentive tokens' value is defined as follows:

$$I_h = c * (P_t + P_r) * t$$

where c is a proportionality constant, P_t is the transmission power used by relay in forwarding the message to the destination and P_r is the power consumed in receiving the message from another source or relay. So, the total incentive promise to a relay is the sum of the incentive due to the software and hardware factors:

$$I = min\{I_s + I_h, I_m\}$$

In addition to the incentive tokens provided by above formula, relay nodes are compensated for additional annotations applied to the in-transit messages. The relay node delivering the message with the added relevant annotations to a destination collects this additional incentive from the destination. If a relay adds n additional keywords and only x are relevant for a destination, the destination will only compensate for x tags. Given the set of relevant added tags T$\{t_1, t_2 .. t_x\}$. The incentive I_t rewarded for added tags is:

$$I_t = min\{\sum I_{t_k}, I_c\}$$

I_c is a cap on incentive tokens for additional tags. I_{t_k} is the incentive due to one added tag defined as follows:

$$I_{t_k} = z * I_m$$

where I_m is the maximum incentive defined earlier and z is a constant such that $0 < z < 1$.

3.3. DISTRIBUTED REPUTATION MODEL

The main motivation behind the development of this model is to reduce the number of malicious users which might add irrelevant tags for a message in the pursuit of acquiring more incentive tokens. For example, it might happen that a message containing an image is tagged with keywords "red car" but there is no "red car" in the image. In such a case, a destination will end up paying for the irrelevant tags as well, thus wasting it's incentive tokens for sharing. To counteract this problem, we propose the following solution. Nodes can assign a rating to messages that they acquire. A recipient node can rate the source and other intermediate nodes(if any) of a message. The source is rated for the message quality and the added tags whereas an intermediate node is rated for any of the added tags in the process of content enrichment. These nodes subsequently calculate the device ratings for the source and relays. Rating of a node is calculated as an average of ratings of the messages received from that node. They share this rating with the next hop in the path of message traversal to a destination. When a message eventually reaches the destination, the delivering device also sends the destination the ratings for the message from all the hops in the path that might have assigned a rating to the message. The destination utilizes these ratings and any previously acquired knowledge about the reputation of the delivering node to decide the number of incentive tokens to be awarded to the delivering node. The reputation of the deliverer is taken into account in order to avoid highly penalizing a good deliverer.

- Rating of a message

When a user receives a message, the user can assign the message rating for the nodes in the path of the message. Source of the message is rated based on the amount of relevant annotations as well as quality of the message as input by the recipient. A relay in the path is rated for the additional annotations that it might have added to the message. A user might not have a total confidence in the rating he/she inputs for the annotations. For example, if an image of a person named "Adam" is annotated with the keyword "Adam" but the user has a conjecture that the image is in fact of a person named "Bill". The user is not entirely certain. In this case, the user can add a confidence value on the ratings of tags. The rating of a source node is calculated as:

$$R_i = 1/2 * (R_t * C/C_m) + 1/2 * R_q$$

The rating of an intermediate node in the path is calculated as:

$$R_i = (R_t * C/C_m)$$

Following this, the rating for nodes in the path of the message is computed.

1. Rating of a node and incentive award

Rating of a node is computed in two cases: 1) When a node in the path of a message assigns a rating to nodes in the path before it, and 2) When say node 1 receives rating of node 2 from node 3. These cases are handled in the following manner:

Case 1: When a node u assigns a rating corresponding to a message M to a node v in the path of M, u calculates the rating of v as an average of ratings of messages received from device v.

$$r_{v,u} = \sum r_{m_v} / N$$

where N is the total number of message received from node v.

Case 2: When node u receives a rating of node v from node z, u performs the calculation for rating of node v as follows:

$$r_{v,u} = (1 - \alpha) * r_{v,z} + \alpha * r_{v,u}$$

Consider a destination node u receiving a message m_v from node v. v also transmits the ratings assigned to all the nodes X corresponding to m_v. Then, the incentive awarded to node v by node u is as follows:

$$I_v = ((1 - \alpha) * (\sum r_{m_{v,x}}) / N + \alpha * r_{v,u} / r_m)) * (I + I_t)$$

where $\alpha > 0.5$, N is the number of hops in the path of the message m_v from a source to the destination u

Summing up everything, the overall data flow between two connected devices u and v is as follows: Devices u and v connect, and share their interests with each other. u generates two sets of messages, a set of messages for which v is a destination and another set of messages for which v is a relay. For the first set, u requests v the promised incentive tokens while also sharing the ratings of the nodes in the path of the message. Device v calculates I_a and checks whether I_a is less than the number of incentive tokens left on it. If it has that many tokens left, it awards the deliverer and the deliverer then delivers the message. The second set of messages from u to v for which u is a relay is further divided into two subsets. The first subset consists of messages for which v has a very high probability of meeting the destination whereas the second set consists of messages for which v has a higher probability of sending the message compared to u. u asks for a percentage of promised incentive from v. If v has that many tokens left, they are awarded to u and the message is received. v receives the full incentive promised by delivering the message. u subsequently forwards the messages in the second subset free of cost to v along with the incentive promised.

4. OPERATOR FUNCTIONS

This section describes the user defined functions that the system must perform in conformance with the proposed approach. Some of these functions require human intervention. The format of the operator functions is given below:

Function 1: Operator Function (Input(optional))

Returns: Output

User Task(Optional): Instructions the crowd needs to follow during operations.

Task: Information regarding what operations (rank, verify, etc.) the crowd/worker has to perform over the event.

As can be seen from above, user task and input are optional entities. For use of some of the functions defined below, a user defined class called Message is created. The format of this message is given earlier.

1) Annotate

This is the task a DTN user performs to annotate a file with the keywords that describe the file. We consider the example of an image file. When a user selects an image file from the file system of his/her device, the system can fetch semantics of keywords from a cloud if the online network connectivity is available. The user can modify some of the labels fetched, keeping the ones that suits the image. The user can also add custom labels of his/her own. For example, the cloud's image processing platform might not be able to recognize face of a person in the image but the user knows who that person is. Thus, the user can add this name to the annotations of the image. Furthermore, all the keywords are assigned a weight of 0.5 initially when the annotations are saved. This initial weight is in accordance with the ChitChat algorithm.

Function 2: Annotate(byte[] ImageFile)

Returns: (String[] keywords)

User Task: Modify the labels fetched if required, assign message priority and save the labels

Task: Fetch and save labels for the image from cloud, save location and timestamp of the image

2) Subscribe

This function lets a user add keyword based interests that act as subscription keywords. Any connected device possessing a message annotated with these subscription keywords can share the messages with this user.

Function 2: Subscribe(String[] interests)

Returns: Void

User Task: Add keywords to the set of interests

Task: Save interests in the database

3) DecayWeights

This function corresponds to the ChitChat's decay algorithm. Weights are decayed upon establishing connection with a device within communication range.

Function 3: DecayWeights(String[] keywords, double[] weights)

Returns: (double[] weights)

Task: Decay weights according to phase one of ChitChat routing weights' exchange algorithm

4) IncrementWeights

This function corresponds to the ChitChat's growth algorithm which decays weights according to the growth model. The resulting output is an incremented weights for the given keywords.

Function 4: IncrementWeights(String[] keywords,double[] weights)

Returns: (double[] weights)

Task: Increment weights according to phase two of ChitChat routing weights' exchange algorithm

5) GetMessagesToForward

This function finds all the messages to be forwarded to a connected device with certain discovered IP address or MAC address, depending upon the mode of communication used, WiFi or Bluetooth.

Function 5:GetMessagesToForward(String[] RemoteDeviceInterests,String DeviceAddress)

Returns: (Message[] messages)

Task: Iterate over the input interests array mapped with the device address and find the corresponding messages

6) DeviceType

After determining a set of messages to be forwarded to a connected device, it is important to decide if the connected device is a destination or a relay for every message. Hence, this function is executed for all the messages intended to be forwarded to aconnected device. A connected device is a relay if the subscription keywords of the device is transient, otherwise it is a destination.

Function 6: DecideDestOrRelay(Message message,String MacAddress)

Returns: (String role)

Task: Determines if the connected node is a destination or a relay

7) BestRelay

Weight of an interest is a measure of encounter probability of the connected device with a destination. When a device u is connected to more than one device with common interests, this function decides the best relay to forward a message based on the weights of interests. Message is forwarded to a relay having the highest encounter probability with the destination. The goal is to reduce the number of messages considerably without significantly affecting the message delivery ratio.

Function 7: DecideBestRelay(String[] MacAddresses,Message message)

Returns: (String MacAddress)

Task: Return the best possible relay to forward the input message

8) Incentivize

When the message sharing module is executed, a source or relay forwarding the message to either a relay or destination executes this function. If a source forwards a message to a destination, the source computes the incentive tokens and requests them from the destination before forwarding the message. If a source or relay is forwarding the message to a relay, they can execute this function to determine the number of tokens that need to be given to the receiving relay.

Function 8: ComputeIncentive(Message message,String MacAddress)

Returns: (double incentiveToken)

Task: Calculate the incentive tokens for this message

9) RateMessage

This function allows a user to rate a message according to DRM. The higher the ratings of the message, higher is the possibility of the average of messages received from the same source.

Function 9: RateMessage(Message M)

Returns: (double Rating)

User Task: Add ratings for i) message quality, ii)keywords for

the message and iii) give confidence value on ratings of keywords

Task: Calculate a rating value for the message

10) RateNode

This function is used to calculate the rating of a device. This function belongs to DRM module. This is calculated by performing the aggregate function average on ratings of the message from the node. The input is device address of a source node in question.

Function 10: RateNode(String MacAddress)
Returns: (double Rating)
Task: Calculate rating for device based on all messages from the device as a source

11) Enrich

This function is used by a relay to add further annotations to a message received from another device and to be relayed.

Function 11: Enrich(Message message, String[] annotations) Returns: (String[] newAnnotations)
Task: Save the added annotations from the user
User task: Add additional relevant keywords to the message

5. PERFORMANCE EVALUATION

In this section, experimental settings are introduced first followed by the results. All the experiments were conducted in ONE simulator version 1.6.0. We compare the performance of our approach with the ChitChat routing algorithm upon which our work is built on, and show how the proposed algorithm improves the performance. The results shown are average of five simulation runs. Note that no other previous incentive scheme considers content enrichment and ranking in DTN, and ChitChat beats other recent competitive DTN social-based routing algorithms, therefore, we improve ChitChat algorithm and our scheme can be integrated with any other DTN routing. The following table shows the parameters used in our experiments.

Table 5.1. Simulation Parameters

Configuration	Default Values
Number of Participants	500
Pool of Social Interest Keywords	200
No of Defined Social Interests	20 per node
Transmission speed	250 kBps
Transmission radius	100 meters
Buffer capacity	250 MB
Message Size	1 MB
Area	5 sq.km.
Simulated time	24 hours
Threshold for relay	0.8
Number of initial tokens	200 per node

In the table above, threshold for relay is explained with an imaginary scenario mentioned subsequently. Consider a scenario in which relay 1 encounters relay 2 and a connection is established between these nodes. Relay 1 possesses a message M for relay 2.M is tagged with a set of keywords T. If the average of weights of tags in T for relay 2 as known by relay 1 upon initial exchange of keywords is greater than the relay threshold, relay 2 pays for a fraction of incentive being promised to it by relay 1 for M. To calculate the performance of our approach, the experiments were conducted to find the effect of fluctuation of selfish nodes' percentage on message delivery ratio and traffic. Variation of percentage of malicious nodes and it's effect on MDR is also checked. Note that all the experiments are conducted under Random Waypoint mobility model. Following subsections explain various conducted experiments in detail:

A. Effect of Selfish Users on Message Delivery Ratio

This is the result shown in Figure 5.1. In this experiment, the number of selfish nodes kept is 500. We vary the percentage of selfish nodes at a rate of 10% from 0 to 100

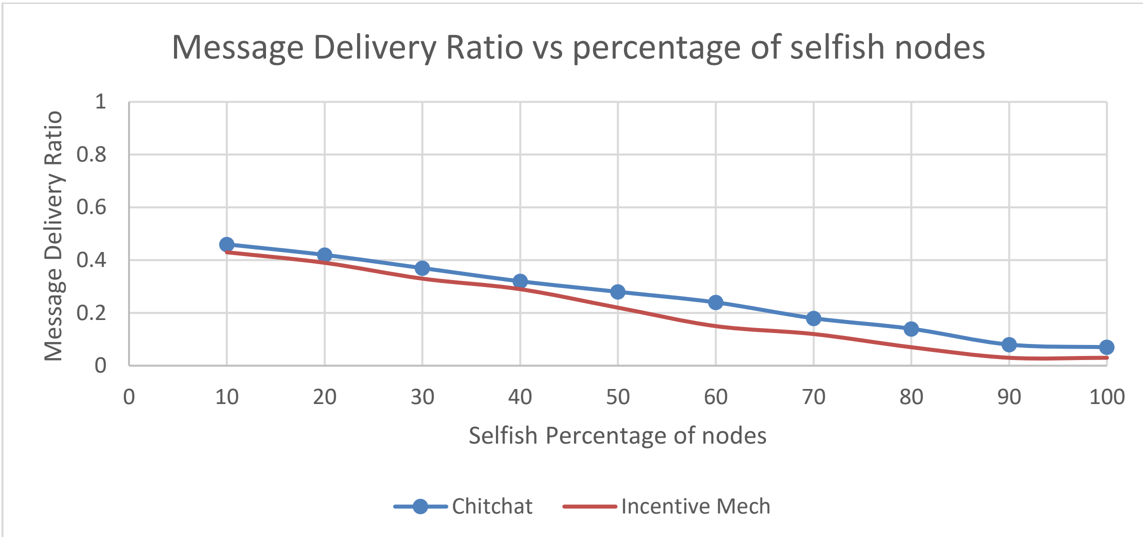

Figure 5.1. MDR vs Percentage of Selfish Nodes

percent. The message delivery ratio in our approach is slightly lesser than ChitChat. The reason behind this is the exhausted incentive tokens in nodes while performing the message disseminations. There is inverse proportionality between the message delivery ratio and percentage of selfish nodes. The primary reason behind this is that as the number of selfish nodes rises, there will be lesser message dissemination because the selfish node randomly participates in forwarding messages. For this experiment, the selfish nodes forward a message to another encountered node in the network one out of ten times. This is simulated by switching off the communication medium in the selfish nodes. A selfish node has its communication medium open one out of ten times when it encounters another node in the network. This is also the reason why the delivery ratio does not drop to absolute zero even when the percentage of selfish nodes is 100.

B. Reduced Traffic compared to ChitChat

This experiment is to find out what percentage of traffic is reduced over ChitChat while there is a reduction in the delivery ratio. The result for this experiment is shown in Figure 5.2 where all the parameters are kept exactly the same as the first experiment. As it can be seen, the higher is the selfish nodes %, more the traffic is reduced. This result makes sense because as the percentage of selfish nodes rises, the incentive tokens assigned to them initially to participate in message forwarding, are exhausted faster.

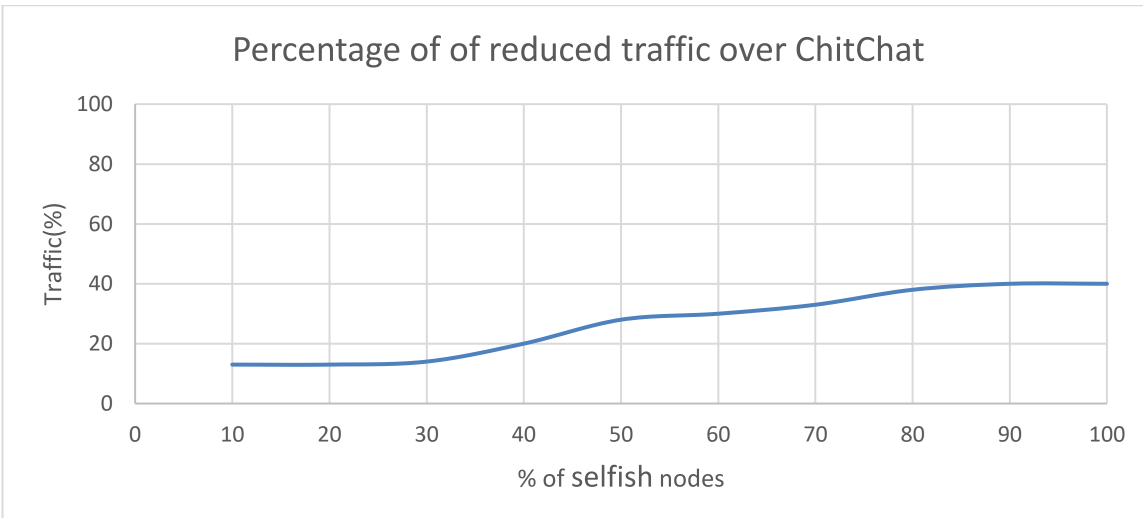

Figure 5.2. Percentage of Reduced Traffic over ChitChat

C. Effect of Tokens on the Message Delivery Ratio

The initial tokens assigned to the nodes are the ones that they use for participation in message forwarding. The result of this experiment is shown in Figure 5.3.

The plots show that for a higher selfish percentage, the message delivery ratio is lower. It also manifests that as the number of tokens assigned to the nodes increases, the message delivery ratio also increases. This is because of the fact that incentive tokens in the nodes will not exhaust rapidly because of high value of initially assigned tokens.

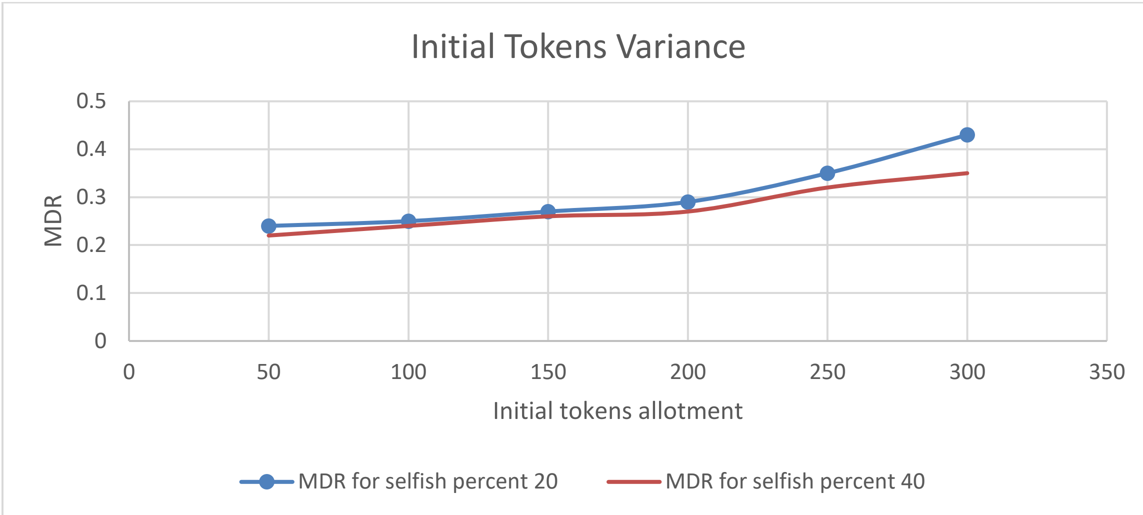

Figure 5.3. Initial Tokens' Variance

D. Recognition of Malicious Nodes

Malicious nodes are defined as the nodes which might either add irrelevant tags to the message or create low quality messages towards the goal of attaining higher incentive

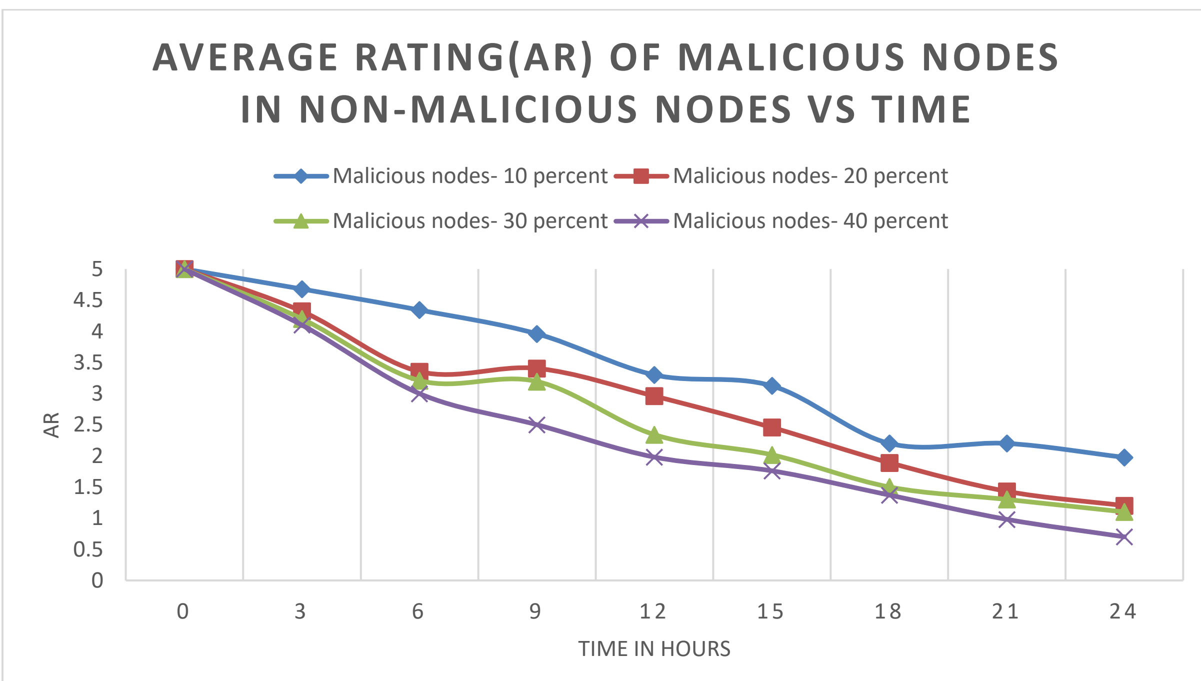

Figure 5.4. Average Rating of Malicious Nodes in Non-Malicious Nodes vs Time

tokens by propagating messages in the network. In Figure 5.4, we find the rate at which malicious nodes in the system are identified by non-malicious nodes. It is important to note that all the above experiments are performed with the content enrichment. However, in those experiments, the percentage of malicious nodes was kept to zero. Average rating of malicious nodes in the non-malicious nodes is a factor which can explain the overall capability of the developed Distributed Reputation Model. With this goal in sight, the malicious nodes percentage is varied from 10 to 40 at an interval of 10. The highest rating a node can assign to another node is 5 for the experiments. The time period of the simulation is 24 hours. It is found that with the evolution of time, the recognition of malicious nodes is accelerated. Moreover, as sthe number of malicious nodes increases, faster the malicious nodes are recognized. This is owing to the fact that more the number of malicious users in an area, more are the chances of a non-malicious node to encounter a malicious node. The encountered non-malicious node then shares the rating of this encountered malicious node with another encountered non-malicious node.

E. Effect of Number of Users on Message Delivery Ratio

This experiment was conducted in order to check how MDR is affected with a varying number of users in a fixed area of 5 sq. km. The result is shown in Figure 5.5. The

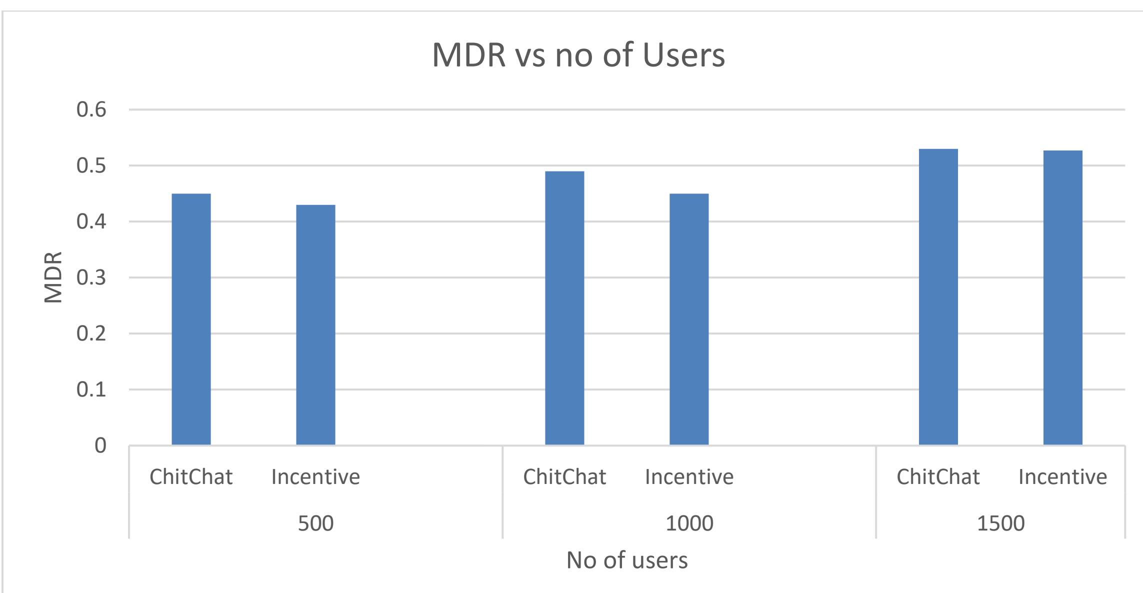

Figure 5.5. MDR vs Number of Users

number of users is varied from 500 to 1500 with an interval of 500 users here. The plot shows that as the number of users increases, the Message Delivery Ratio for both ChitChat and Incentive increases. Another important observation here is that the difference between the MDR for ChitChat and Incentive mechanism decreases gradually with the incremented number of users and this difference almost fades away when the number of users are 1500. The reason behind this gradual decline of MDR difference is that the number of message carriers grows with the rise in the number of users.Therefore, a message might travel through multiple paths.

F. Priority Segmented MDR

In the first two experiments, a decrease in MDR was compensated by the reduced traffic overhead. However, it is important to analyze the quality of messages disseminated

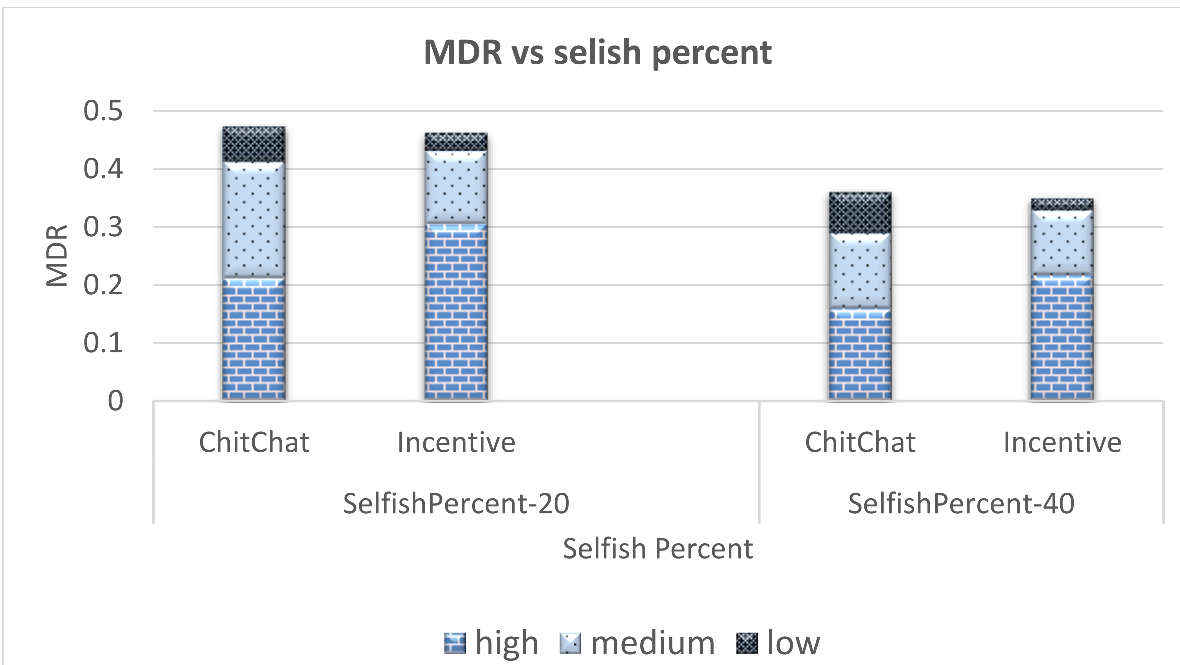

Figure 5.6. Priority Segmented MDR vs Selfish Percent of Nodes

by our scheme. Thus, we ran simulations where selfish percentage of nodes is kept at 20 percent and 40 percent. The number of nodes is 500 in a 5 sq km area. Out of these nodes, 50% of the nodes generated high quality larger size and high priority messages, 30% created medium quality and the rest produced low quality. From the plot, it can be seen that in both the cases, viz., 20% and 40% selfish nodes, higher number of high priority messages is delivered in our scheme compared to Chitchat. This is because our approach prioritizes messages based on the quality as well as the assigned priority. Moreover, the higher quality message has a larger size also, and thus, higher energy is consumed in propagating, and therefore, the incentive received for forwarding such messages is also higher. Therefore, nodes earn higher incentives making it an overall higher priority message dissemination scheme.

6. CONCLUSION AND FUTURE WORK

In this paper, we present a novel content-centric data dissemination technique using a combination of credit as well as reputation based incentive mechanism designed, which is integrated with Chitchat routing [5]. Our proposed scheme successfully delivers higher quality and priority messages while motivating the selfish nodes to participate in content enrichment and message forwarding to earn incentives. Our scheme is also successful in identifying and barring the malicious nodes in the network from receiving messages. Our experiments show that message delivery ratio attained is almost the same as ChitChat while curbing the network congestion and malicious nodes even further. Moreover, the content enrichment makes the content richer as the message propagates deeper into the network. A demo application for our mechanism has also been developed and tested for Android environment for Bluetooth as a communication medium [24]. Stress testing in the real world environment will help in evaluating the performance and usability of our mechanism.

REFERENCES

[1] Sobin, C. C., et al. "A survey of routing and data dissemination in delay tolerant networks." Journal of Network and Computer Applications 67 (2016): 128-146.

[2] Pan Hui, J. Crowcroft, and E. Yoneki. Bubble rap: Social-based forwarding in delay-tolerant networks. *IEEE Transactions on Mobile Computing*, 10(11):1576–1589, Nov 2011.

[3] A. Mei, G. Morabito, P. Santi, and J. Stefa. Social-aware stateless routing in pocket switched networks. *IEEE Transactions on Parallel and Distributed Systems*, 26:252–261, Jan 2015.

[4] Amin Vahdat and David Becker. Epidemic routing for partiallyconnected ad hoc networks. Technical report, Duke University, 2000.

[5] McGeehan, Douglas, Dan Lin, and Sanjay Madria. "ChitChat: An Effective Message Delivery Method in Sparse Pocket-Switched Networks." 36th IEEE International Conference on Distributed Computing Systems (ICDCS), 2016.

[6] Tatiana Seregina, Olivier Brun, Rachid Elazouzi, Balakrishna Prabhu. On the Design of a Reward-Based Incentive Mechanism for Delay Tolerant Networks. IEEE Transactions on Mobile Computing, Institute of Electrical and Electronics Engineers, 2017, 16 (2), pp.453-465.

[7] K. Sugiyama T. Kubo A. Tagami A. Parekh "Incentive mechanism for DTN-based message delivery services", in Proc. IEEE GLOBECOM, pp. 3108-3113 Dec. 2013.

[8] Rachid El-Azouzi, Francesco De Pellegrini, Habib B.A. Sidi, and Vijay Kamble. Evolutionary forwarding games in delay tolerant networks: Equilibria, mechanism design and stochastic approximation. Computer Networks, 2012.

[9] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance modeling of epidemic routing. Computer Networks, 51:2867-2891, 2007.

[10] U. Shevade, H.H. Song, L. Qiu, and Y. Zhang. Incentive-aware routing in dtns. In IEEE International Conference on Network Protocols (ICNP), pages 238-247, 2008.

[11] B.B. Chen and M.C. Chan. Mobicent: a credit-based incentive system for disruption tolerant network. In Proc. of IEEE INFOCOM, pages 1-9, 2010.

[12] R. Lu, X. Lin, H. Zhu, X. Shen, and B. Preiss. Pi: A practical incentive protocol for delay tolerant networks. IEEE Transactions on Wireless Communications, 9(4):1483-1493, 2010.

[13] H. Zhu, X. Lin, R. Lu, Y. Fan, and X. Shen. Smart: A secure multilayer credit-based incentive scheme for delay-tolerant networks. IEEE Transactions on Vehicular Technology, 58(8):4628-4639, 2009.

[14] L. Wei, Z. Cao, and H. Zhu. Mobigame: A user-centric reputation based incentive protocol for delay/disruption tolerant networks. In Proc. IEEE Global Telecommunications Conference (GLOBECOM), pages 1-5, 2011.

[15] Qinghua Li, Sencun Zhu, and Guohong Cao. Routing in socially selfish delay tolerant networks. In Proc. of IEEE INFOCOM, pages 1-9, 2010.

[16] Ting Ning, Zhipeng Yang, Xiaojuan Xie, and Hongyi Wu. Incentive-aware data dissemination in delay-tolerant mobile networks. In Proc. of IEEE Sensor, Mesh and Ad Hoc Communications and Networks (SECON), pages 539-547, 2011.

[17] Md Yusuf Sarwar Uddin, Brighten Godfrey, and Tarek Abdelzaher. Relics: In-network realization of incentives to combat selfishness in DTNs. In Proc. IEEE International Conference on Network Protocols (ICNP), pages 203-212, 2010.

[18] Yan Wang, Mooi-Choo Chuah, and Yingying Chen. Incentive driven information sharing in delay tolerant mobile networks. In Proc. of IEEE Global Communications Conference (GLOBECOM), pages 5279-5284, 2012.

[19] W. Chahin, H. Sidi, R. El Azouzi, F. De Pellegrini, and J. Walrand. Incentive mechanisms based on minority games in heterogeneous delay tolerant networks. In Proc. of The International Teletraffic Congress (ITC), 2013.

[20] Esteban Moro. The Minority Game: an introductory guide. eprint arXiv:cond-mat/0402651, February 2004.

[21] Wenjie Li, Laura Galluccio, Francesca Bassi, Michel Kieffer. Distributed Faulty Node Detection in Delay Tolerant Networks: Design and Analysis. 2016. <hal-01327472>

[22] Kravari, Kalliopi, and Nick Bassiliades. "DISARM: A social distributed agent reputation model based on defeasible logic." Journal of Systems and Software 117 (2016): 130-152.

[23] Ari Ker¨anen, J¨org Ott, and Teemu K¨arkk¨ainen. The ONE Simulator for DTN Protocol Evaluation. In Proceedings of the 2nd International Conference on Simulation Tools and Techniques, SIMUTools ’09, New York, NY, USA, 2009. ICST.

[24] Jethawa, Himanshu, and Sanjay Madria. "Incentive Mechanism for Data-Centric Message Delivery in Delay Tolerant Networks." Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on. IEEE, 2017.

[25] Magaia, Naercio, Paulo Pereira, and Miguel Correia. "REPSYS: A Robust and Distributed Reputation System for Delay-Tolerant Networks." (2017).

[26] Li, Na, and Sajal K. Das. "A trust-based framework for data forwarding in opportunistic networks." Ad Hoc Networks 11.4 (2013): 1497-1509.

[27] Ayday, Erman, and Faramarz Fekri. "An iterative algorithm for trust management and adversary detection for delay-tolerant networks." IEEE Transactions on Mobile Computing 11.9 (2012): 1514-1531.

[28] Dias, Joao AFF, et al. "A cooperative watchdog system to detect misbehavior nodes in vehicular delay-tolerant networks." IEEE Transactions on Industrial Electronics 62.12 (2015): 7929-7937.

[29] https://en.wikipedia.org/wiki/Friis_transmission_equation

II. INCENTIVE MECHANISM FOR DATA-CENTRIC MESSAGE DELIVERY IN DELAY TOLERANT NETWORKS

ABSTRACT

A key issue in delay tolerant networks (DTN) is to find the right node to store and relay messages. We consider messages annotated with the unique keywords describing the message subject, and nodes also adds keywords to describe their mission interests, priority and their transient social relationship (TSR). To offset resource costs, an incentive mechanism is developed over transient social relationships which enrich en-route message content and motivate better semantically related nodes to carry and forward messages. The incentive mechanism ensures avoidance of congestion due to uncooperative or selfish behavior of nodes.

1. INTRODUCTION

Delay Tolerant Networks (DTN) use store and forward paradigm for message dissemination in an environment that lacks continuous connectivity due to the presence of obstacles/inactive nodes or in absence of communication infrastructure support. In DTN, users can share messages based on their social interests. For a higher data delivery rate, we use message annotations in [2] for in-transit mission packets for data-centric intelligent routing decisions. A node passes messages to intermediate individuals, which match or exceed the strength of keyword-based annotations of the message. In addition, intermediate nodes add keywords-based annotations to create higher content strength messages and will direct messages toward the destination. In our work here, thus, content gets enriched as the situation evolves, such as in a disaster situation, traffic event/condition or in a battlefield. However, battery and memory are limited in mobile devices, therefore, even with the most efficient routing algorithms in place, some nodes turn uncooperative to conserve resources. In addition, some nodes may start generating a lot of messages which might lead to unfairness in message dissemination and can cause network congestion.

To counteract these challenges, an incentive mechanism [3] is developed on top of Chit-chat algorithm [1] for better message dissemination and to address selfish behavior. In this approach, incentives are provided to nodes for active participation of the high priority messages forwarding and for enriching the content of en-route messages. The objective of this demo paper is to demonstrate the developed DTN application to address the aforementioned problems.

2. PROBLEM STATEMENT

Given a mobile device possessing a set of messages to be forwarded in DTN, the problem is to design an incentive mechanism on top of our existing ChitChat algorithm [1] for efficient message delivery. The incentive formulation must correctly consider the significant factors that encourage data-centric dissemination of messages. If an interested device does not have any incentive to allocate to a relay node, it should not receive the message for forwarding.

3. ARCHITECTURE

The incentive mechanism developed on top of an already existing ChitChat algorithm [1] achieves faster and higher message delivery ratio using nodes' transient social interests for data delivery. A universal message format is used throughout the network for the sake of consistency. The following subsections illustrate message format, ChitChat algorithm and the incentive mechanism.

3.1. MESSAGE FORMAT

The multimedia message format is shown in Figure 3.1 where (i) topic implies interests (ii) location parameters such as latitude and longitude are stored as keyvalue attributes. A message in the system is an encapsulation of the multimedia data along with some metadata tags. Additionally, a UUID which is a unique identifier for the message makes sure that the message does not get duplicated in any device. A timestamp, the time at which the multimedia content for the message created is also added as a part of the message. Source and destinations, MIME and format of the message are also added.

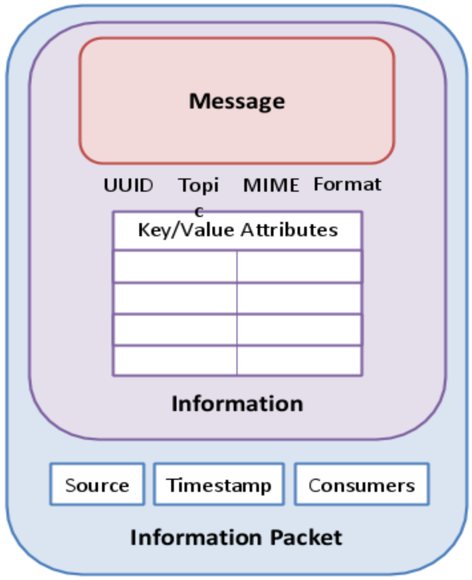

Figure 3.1. Message Format

3.2. CHITCHAT

In ChitChat, social interests can either be created or acquired. They are termed as direct and transient interests respectively. The following sections explain the two phases of ChitChat:

1. Weight exchange phase

For a given node, when a remote node falls in communication range, a connection attempt is initiated. Once the nodes are connected, the weight update algorithm is triggered for every pair of nodes. This algorithm works in three phases. Initially, the weights are decayed as per the decay model. Decayed weights are then shared and weights are finally updated as per growth model. As a result of this, interests of the connected devices can be acquired. The value of weight for acquired interests start to decline as per decay model when no other device with that interest is currently connected. Such interests are therefore only transient in nature.

2. Message routing phase

Once the interests and weights are updated and shared, ChitChat's routing algorithm determines the set of messages that need to be forwarded. The messages are forwarded to all those nodes which are either destinations or a subset of relays which have higher probability of delivering them to the destination, either directly or indirectly. The probability of a message getting delivered is a function of the sum of weights of individual interests mapped with the message. The proposed incentive algorithm is run on forwarding the messages within ChitChat to address uncooperative and selfish behavior. The details of this algorithm are explained in the following section.

3.3. INCENTIVE MECHANISM

The incentive mechanism begins with determining whether the connected node is an intermediate node or a destination.

Based on whether the node has a direct social interest or transient social interest in the message, the node is classified as a relay or destination. If it is an intermediate node, an incentive calculation is done before forwarding the message. Subsequently, the message is forwarded to the connected node along with the promised value of reward equal to the calculated incentive. If the connected node is a destination, the message is forwarded and the incentive reward equal to the promise is collected.

The mechanism works in such a way that an intermediate node is ensured an incentive it can receive when it forwards (may also add additional keywords) a message to the destination. The destination provides the promised incentive. Only a device which first delivers a message to the destination is given the incentive for that message. However, the devices can share a message with multiple destinations.

The value of the promised incentive depends on factors such as the level of interest of the connected node in the content of the message, priority level of the message set by the source, size of the message, quality of the message and energy consumption of the connected devices. Some of these parameters are static whereas others are dynamic. Priority level of the message set by the source, size of the message and quality of the message are static, implying that once the message has been created, they do not change.

Interest level in message and energy consumption are dynamic as they are device dependents.

A value of maximum incentive is predefined. A formula is used to calculate the value of promised incentive as a function of above parameters. The formula also ensures that the incentive promise for a message cannot exceed more than the maximum allowed value. Therefore, an incentive value for a message is either equal to or a fraction of the maximum incentive allowed. Figure 3.2 shows the data flow between two connected nodes.

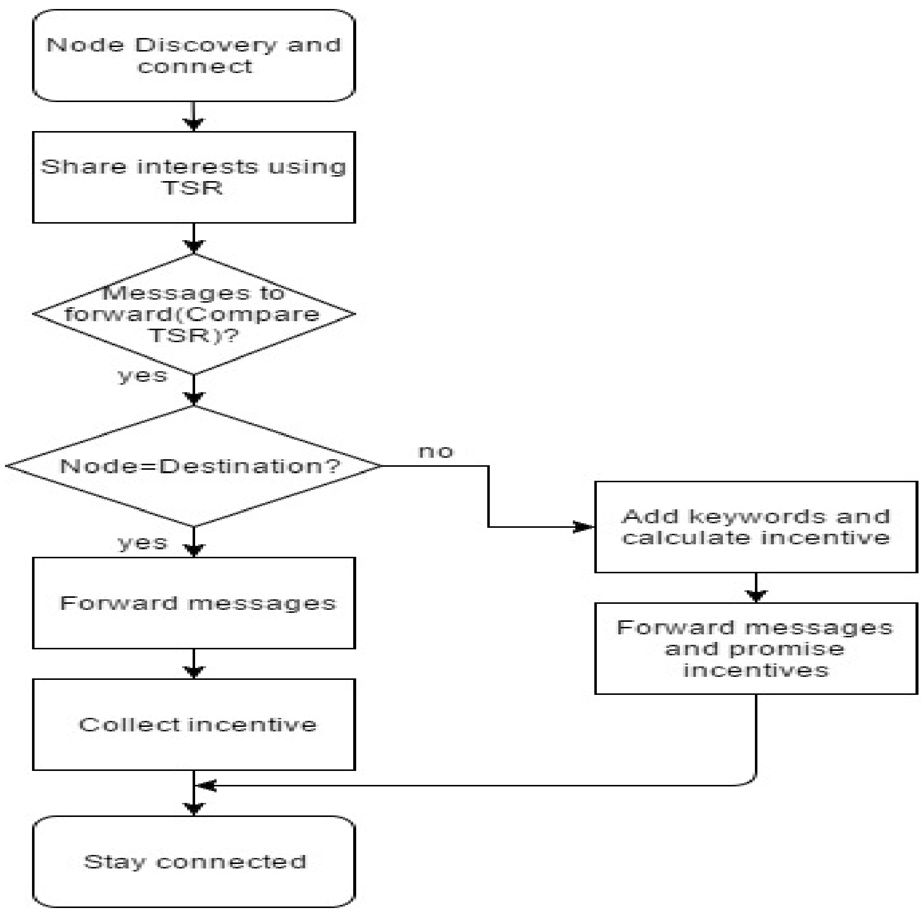

Figure 3.2. Data Flow between Two Nodes

Source generates a message and forwards it to the destinations and relays. Initially, all the devices are assigned the same initial value of incentive tokens before the data flow takes place. The devices can then utilize the allotted tokens to share content. A device with no incentive to offer cannot act as a destination. This implies that if the device has not participated as an intermediary and is eventually left with zero incentive tokens to offer, it has to first participate as an intermediary to gain incentives.

4. DTN IMPLEMENTATION

The DTN application is designed to use Bluetooth communication in Android devices for sharing messages.

Figure 4.1 shows a gallery screen where an image is already selected. A user of the application can generate a message by either clicking an image using camera or select an image from the gallery. In both cases, the user is provided with a set of keywords extracted from the image using Google Cloud Vision API. Following screens

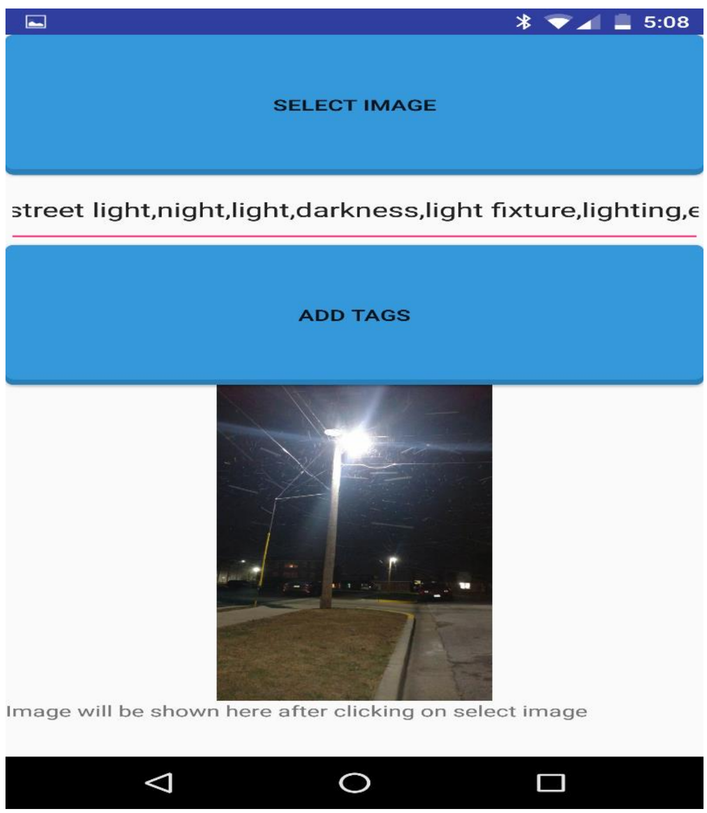

Figure 4.1. Gallery

are shown to the user. After the image is selected, the textbox gets automatically populated with the keywords by Google Cloud Vision API. A user can now edit the keywords and add more. Henceforth, these keywords act as interests for the user. When these keywords are saved using add tags button, a new message is created in the background which saves the added interests, the location where the keywords were edited/added and the timestamp of the image. Alternatively, the user could also click an image using camera screen. The flow in that case would be the same.

Figure 4.2 shows the interests page. This page shows the user-interests using keywords, their weights and the MAC address of the device via which the device acquired the social interest. A value of SELF implies that the interest in the keyword is direct, otherwise it is indirect and hence, transient.

Figure 4.3 shows a screen where two lists are displayed for nearby devices and connected devices. Two other devices were kept in communication range of the application. The screen shows that both the devices are nearby and also connected. There

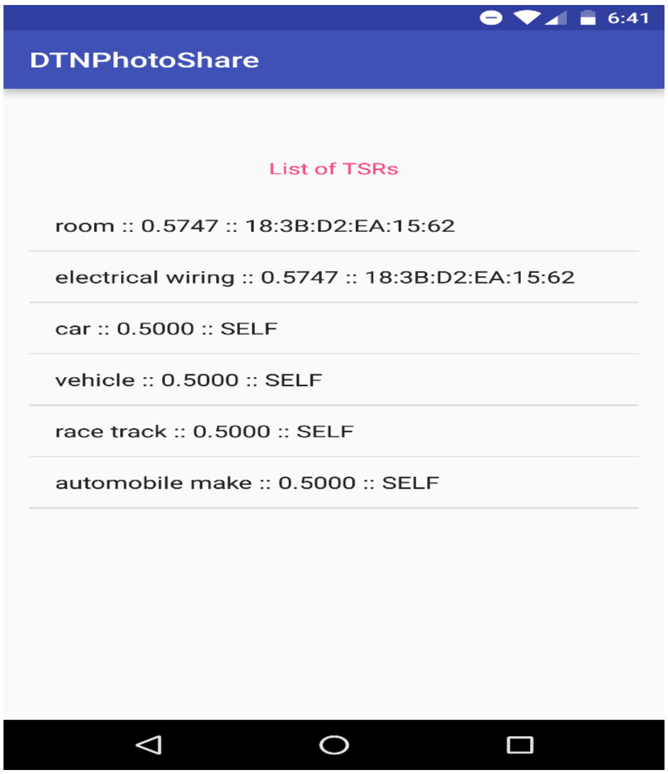

Figure 4.2. User Interests

might be other devices in range but they might not have the developed application installed. Such devices will be shown in nearby devices list but not under the list of connected devices.

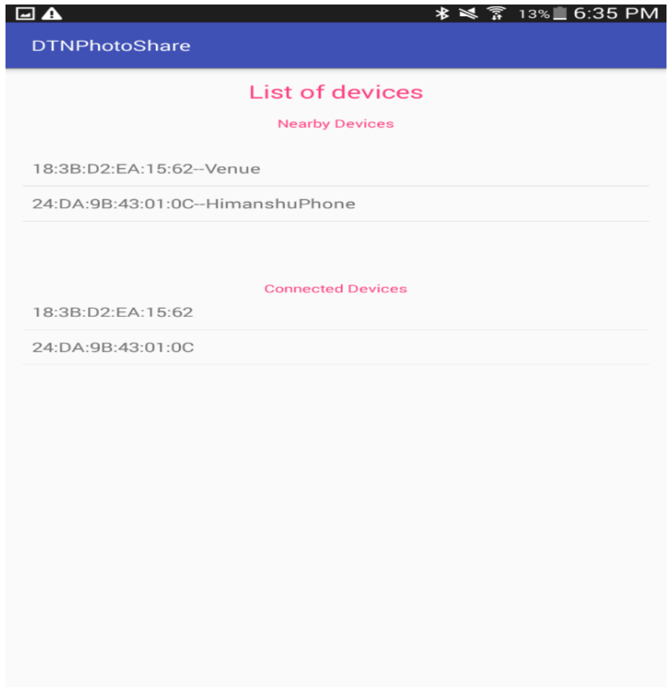

Figure 4.3. Neighbors Listing

Figure 4.4 shows a screen where a list of received messages are displayed in a grid fashion. The images have caption as the path of the image. When a user clicks on any message, a message details screen is shown. The messages received were either destined for the device or to be forwarded so that it can relay the messages. This is determined by clicking on the message which directs the user to the message details screen.

Figure 4.4. Received Messages

Figure 4.5 shows a received message displayed as a combination of an image and it's metadata tags such as Source MAC, Source Name, interests shown as keywords, latitude, longitude, and timestamp. Source MAC and source name identify the originator of the message in the network. Timestamp lets the destination node know when the message was generated.

The application also has screens for checking the available incentives to offer and also to check the messages generated locally. There is also a screen that can be used to add interests manually. The operations mentioned in the architecture always run in the background. Every few minutes, neighboring nodes can be discovered by performing Bluetooth discovery. The energy overhead involved here cannot be eliminated as if a device has no clue to which devices are in the vicinity, it cannot initiate a connection attempt. Also, Bluetooth discovery consumes maximum bandwidth available to a device.

This implies that if a device is already conducting transactions of data transfer, performing Bluetooth discovery may break the existing connections.

Figure 4.5. Message Details

To conquer this hindrance, a solution is devised which first checks if there are any transactions in process at the time discovery is scheduled. If this is true, Bluetooth discovery is delayed until the transactions are finished.

5.DEMO SHOWN

For a demo, three Android devices are taken, viz., devices A, B and C. This is to make sure that there are at least some messages that travel two hops. Initially, all the devices are given an incentive of 50 tokens. Device A is placed in the communication range of device B, whereas device B is kept in communication range of the device C.

However, device A's and device C's Bluetooth range do not overlap. Bluetooth of devices A and B is now switched on. Initially, Device A is stored with 40 messages of varying sizes that device B is interested in. Bluetooth of devices A and B is switched on. When checked, B had received about 22 messages in total based on TSR values (messages whose TSR values were higher at B). Incentives are distributed next based on several factors such as TSR, energy used, priority etc. When the incentive screen is opened in Device B to check the incentives left, it is found that device B has zero reward to offer. Therefore, device B did not receive anymore messages from device A as it does not have any more incentives to offer.

The interests of devices B and C are also kept exactly the same. Next, Bluetooth of device A is switched off and Bluetooth of C is switched on. B now shares the messages with C after it adds some more keywords to make these messages enriched in content. Device B has now obtained some incentives from C. Device A's Bluetooth is turned on again. The remaining 18 messages are transferred to device B as it has now earned incentives. This demo shows that when B had no incentive value, it could not receive any more interesting messages. When C received 18 messages from B, it also adds additional keywords in some of the messages to enrich the message content.

6. CONCLUSION AND FUTURE WORK

This paper demonstrates the developed Android application implementing the proposed incentive mechanism for sharing data using ChitChat in DTN. An incentive mechanism has been implemented which prevents users from becoming selfish and motivates them to relay more content-rich messages. The application also bars selfish users from receiving data intended for them until they have enough incentives to pay for the messages. Currently the application is developed for Bluetooth and Android, and next it will be developed using WiFi Direct which is ad hoc wireless connection. It will also be developed for other operating systems.

Acknowledgement: We are thankful to Doug McGeehan, Nathan Holzhauer, Brian Holmes and Jim Milligan for their comments and help.

REFERENCES

[1] McGeehan, Douglas, Dan Lin, and Sanjay Madria. "ChitChat: An Effective Message Delivery Method in Sparse Pocket-Switched Networks." 36th IEEE International Conference on Distributed Computing Systems (ICDCS), 2016.

[2] Roy Cabaniss, SS Vulli, Sanjay Madria, Social group detection based routing in delay tolerant networks, Wireless networks 19 (8), 1979-1993, 2013.

[3] Anil Jade, Sanjay Madria, Mark Linderman, Incentive based routing protocol for mobile peer to peer networks, in 10th IEEE International Conference on Mobile Data Management (MDM), 2009, Taiwan.

SECTION

3. CONCLUSION

We have developed an incentive mechanism for data-centric file sharing in Delay-Tolerant networks. We have observed that the Message Delivery Ratio(MDR) is slightly lower compared to Chitchat for not too high value of initial tokens. The results lead us to conclude that the slightly lesser MDR is compensated by the highly diminished traffic. More higher quality and higher priority messages are delivered in our incentive approach. To achieve higher MDR, assign higher initial number of tokens to all the nodes but in this case, the traffic reduction is also moderate. Depending upon the application, the number of initial tokens can be set considering the trade-off between the Message Delivery Ratio and the traffic. The demo of the application shows the usability of the developed scheme in real world scenario.

REFERENCES

[1] Neves dos Santos, Francisco, et al. "Cedo: Content-centric dissemination algorithm for delay-tolerant networks." *Proceedings of the 16th ACM international conference on Modeling, analysis & simulation of wireless and mobile systems*. ACM, 2013.

[2] McGeehan, Douglas, Dan Lin, and Sanjay Madria. "ChitChat: An Effective Message Delivery Method in Sparse Pocket-Switched Networks." 36th IEEE International Conference on Distributed Computing Systems (ICDCS), 2016.

[3] B.B. Chen and M.C. Chan. Mobicent: a credit-based incentive system for disruption tolerant network. In Proc. of IEEE INFOCOM, pages 1-9, 2010.

[4] R. Lu, X. Lin, H. Zhu, X. Shen, and B. Preiss. Pi: A practical incentive protocol for delay tolerant networks. IEEE Transactions on Wireless Communications, 9(4):1483-1493, 2010.

[5] Tatiana Seregina, Olivier Brun, Rachid Elazouzi, Balakrishna Prabhu. On the Design of a Reward-Based Incentive Mechanism for Delay Tolerant Networks. IEEE Transactions on Mobile Computing, Institute of Electrical and Electronics Engineers, 2017, 16 (2), pp.453-465.

[6] Yan Wang, Mooi-Choo Chuah, and Yingying Chen. Incentive driven information sharing in delay tolerant mobile networks. In Proc. of IEEE Global Communications Conference (GLOBECOM), pages 5279-5284, 2012.

[7] U. Shevade, H.H. Song, L. Qiu, and Y. Zhang. Incentive-aware routing in dtns. In IEEE International Conference on Network Protocols (ICNP), pages 38-47, 2008.

[8] Magaia, Naercio, Paulo Pereira, and Miguel Correia. "REPSYS: A Robust and Distributed Reputation System for Delay-Tolerant Networks." (2017).

[9] Li, Na, and Sajal K. Das. "A trust-based framework for data forwarding in opportunistic networks." Ad Hoc Networks 11.4 (2013): 1497-1509.

[10] Ayday, Erman, and Faramarz Fekri. "An iterative algorithm for trust management and adversary detection for delay-tolerant networks." IEEE Transactions on Mobile Computing 11.9 (2012): 1514-1531.

[11] Dias, Joao AFF, et al. "A cooperative watchdog system to detect misbehavior nodes in vehicular delay-tolerant networks." IEEE Transactions on Industrial Electronics 62.12 (2015): 7929-7937.

VITA

Himanshu Jethawa was born in Pusad, India. He received distinction in Bachelor of Engineering degree in Computer Science and Engineering from Mumbai University, India in 2013. After working at a startup in Mumbai for two years, he became a graduate student in the Computer Science Department at Missouri University of Science and Technology in August 2015 and worked as a Graduate Research Assistant under Dr. Sanjay Kumar Madria from January 2016 to December 2017. He received his Master of Science in Computer Science at Missouri University of Science and Technology in May 2018.