Wireless Pers Commun
DOI 10.1007/s11277-016-3277-x

Analysis of Creido Enhanced Chord Overlay Protocol Under Different Movement Models in Delay Tolerant Networks

M. Umaparvathi[1] · N. Bhalaji[2] · S. Jothi Prasanna[3] · N. Parthiban[4]

Abstract Delay Tolerant Network (DTN) is an enlivening area to delve in owing to its rudimentary characteristics like intermittent links, variable delays etc. and its physiognomies of applications wherever traditional networks like sparse and emergency mobile ad-hoc networks fails. These applications are aided extensively if the performance metrics are recuperated. In this article a novel plot to increase the end to end bundle delivery ratio by assimilating CREIDO bundles in chord overlay network. This periodic bundle invokes the stabilize function of chord overlay protocol to make the overlay topology robust during node join and exits. In this article, OverSim framework of OMNeT++ is used under different mobility models to analyse the performance metrics of CREIDO bundles. Thus the results obtained were encouraging and outsourced the existing protocol in aspects of Bundle Delivery Ratio and throughput.

Keywords Delay tolerant networks · CREIDO · Chord overlay network · Chord protocol · Movement models

✉ M. Umaparvathi
umaparvathicbe@gmail.com

1 Department of ECE, RVS College of Engineering and Technology, Coimbatore, India

2 Department of Information Technology, SSN College of Engineering, Chennai, Tamil Nadu 603 110, India

3 Department of Computer Science and Engineering, SRM University, Chennai 603203, India

4 VIT University, Chennai 600 127, India

Published online: 09 April 2016

1 Introduction

DTN descends from Inter Planetary internet [1]. DTN resemble Mobile Ad hoc Networks (MANET) but contretemps in number of mobile nodes subjugated per unit area. The rudimentary characteristics of DTN encompasses tolerating long delay and distance communication in the dearth of centralized device [2]. It also stipulates end-to-end service which endures intermittent connectivity, asymmetric and symmetric links and variable delays. Bundle reaches the destination through the mobile nodes depending on other mobile node that acts as relay in the presence of aforesaid challenges. This forwarding strategy is known as Opportunistic routing [3, 4]. The DTN protocol includes the ability to cope up with the environment, capability to handle scheduled and opportunistic connectivity, custody transfer and security [5].

1.1 Overlay Networks

An overlay Network is the virtual topology on the physical topology which acts as an interface to the users (see Fig. 1) [6]. These are the virtual networks which are logically built on the top of the physical network by ordering the nodes structurally or without any structure to enhance the data communication [7]. These have gained momentum in the recent years as it overcomes several deployment barriers of the decentralized networks such as multicasting, inter domain routing pathologies, content distribution and content sharing. The hosts in the overlays multicasts exclusively deal with group management and routing without seeking any help from other internet routers [8].

- An overlay allows both protocol developers and application users to design and instigate their own protocols and the network environment on the internet.
- Overlay networks endow with data routing which are flexible, speedily detectable and eludes congestion in networks by opting the paths adaptively according to the desired and required metrics such as probed latency.
- Scalability and Robustness are the two main features which are highly supported by overlays. The end nodes are highly connected due to bendable routing. This high connectivity of end nodes facilitates effectual sharing of huge amount of data spread across the internet.

1.2 Structured Overlay Networks

Structured Overlay Networks uses some form of NodeID's to construct a structure through arrangement of nodes and this structural arrangement in the overlay topology is the key feature which enhances numerous services offered in overlays. The above discussed

VERSION (1 BYTE)	FLAGS (1 BYTE)	SOURCE ID (160 BITS)	SOURCE IP (32 BITS)	DEST. ID (160 BITS)	DEST. IP (32 BITS)	RESERVED (2 BYTES)

Fig. 1 CREIDO bundle specification

concept is acknowledged as key based routing (KBR) [9]. Distributed Hash Tables (DHT) [10] determines the responsibility of nodes which are dependent on their NodeID's for data storage. KBR and DHT depends on identifier range, with respect to a node's own ID. With the given distance metric [8], a node is responsible for locating an identifier if its NodeID is closest to this identifier with respect to the given metric. Structured Overlay Networks allows efficient implementation of DHT with O(log N) as lookup time where N is the number of nodes in the overlays [8].

1.3 Chord Overlay Protocol

Chord is a ring like structure [11]. This is an overlay protocol that coordinates the nodes in clockwise directed circle with respect to their 128 bit NodeID's. The chord utilizes linear distance as a metric for arrangement of nodes. NodeID for each node is computed by hashing the corresponding IP address of that device [11]. Each node in chord ring has a successor and a predecessor which are numerically contiguous in the identifier space [12]. Finger tables are preserved in the chord overlay protocol for the purpose of accelerating lookups and routing [13]. In the chord overlay network, further connections to the nodes are in logarithmic distance. When a message has to be sent to Node ID A, the message is sent to the farthest finger with the preceding ID. The node where the finger points to is the better node as it is closer to the destined node. With this procedure, Chord can deliver the bundles in O(log N) steps, with N being the number of nodes in the overlay. For the adept functioning of structured Overlay Networks, the nodes in the overlay should have random distribution of Node ID's [12, 14].

1.4 CREIDO Bundle

CREIDO bundle projected in this article is the periodic bundle that makes the architecture of chord overlay protocol more robust and increases the reliability measure of reaching the destined node. This is a periodic bundle which is sent by the nodes in the overlays to their neighbors. This bundle invokes the *stabilize* function [15] instantly during node joins and departures from the network without waiting for the periodic run of stabilize protocol. Henceforth, the bundle delivery ratio will be improved.

2 Literature Review

Chord is a DHT based resource routing protocol [13, 16–18]. Using consistent hashing technique, it allocates each node and resource an m-bit identifier. The resource and nodes are orchestrated to a logical ring known as chord ring. There are two problems in the prevailing chord overlay protocols as stated by authors in [19, 20], One is the redundant information present in the finger table of each node. Second is use of consistent hashing technique to build the chord ring due to which the logical and physical address may vary greatly making the lookup a high stint phenomenon [19, 20]. Researchers have devised some methods to overcome the above stated conundrums so that the performance of the chord overlay protocol is recuperated. In [20], researchers have devised an algorithm to remove redundant information in the finger table with the addition of counter clockwise node information holding good record in time and space complexity. In [21], authors have devised a bilateral chord that can also act in counter clockwise direction. This improves the performance but occupies more space. The high chord search algorithm improves the distribution of routing table pointer density but this method has to maintain more routing tables [21].

The method of self-stabilization is proposed by Dijkstra [22] in the field of distributed computing. In [24], authors suggest that self-stabilization is the general approach to improve fault tolerance in overlay and p2p networks. With the imminent p2p and overlay protocols, this method of self-stabilization is experimented for the upcoming dynamic p2p and overlay protocols leading to a self-stabilizing p2p network maintenance algorithm that supports logarithmic updates and searches known as Tiara [23]. In [23], authors theoretically presents chord as the possible extension of Tiara to a ring structure but it is not elaborated and experimented. In [25], authors present 3nuts, a self-stabilizing p2p network supporting large range queries and acclimating to the underlying physical network. This protocol combines random networks, prefix trees and DHTs. This protocol measures Round Trip Delay to adapt overlay to the underlay networks. But the major setback is higher degree robustness is not possible in case of highly skewed data distribution.

In [26], authors described a technique called linearization to transmute any capriciously connected network into a sorted list. But, the method of self-stabilization is not introduced in [26]. The extension of this technique is presented in [27] as linearized De Bruijn network by solidifying the shortfall of [26] by the discretization of a continuous variant of the classical De Bruijn. This tries to be fault tolerant by maintaining a constant node degree. The major shortfall of [27] is the higher time complexity. Re-chord [28] which uses linearization technique and local information uprights any fallacious but weakly connected graphs into the state of Chord. The Re-chord has the ability of self-stabilizing but it is prone to strong churn [28]. Ca–Re-Chord [28] is the extension of the Re-chord that is capable of handling churn conditions well than the Re-chord. Even though this method is resistant to high churn rates, it is experimentally verified that the amount of successful lookups is low than the traditional chord as the number of hops to reach the destination increases with the increase in churn. HP-Chord [29] describes a way to improve routing efficiency by exploiting the bandwidth heterogeneity of nodes and the proximity of the underlying nodes. The main idea here is the creation of virtual nodes. If a new node joins the network, virtual nodes are created along with the new node. Even though this method improves the routing efficiency, the space complexity for holding the virtual nodes is an additional burden. Also, the routing process will be sufficiently complicated leading to a maximum usage of the processor and faster exhaustion of battery powered nodes as there are at least one virtual node for a node based on its bandwidth.

In this article, we propose a periodic bundle termed as CREIDO bundle to find the nodes joining and leaving the network in order to evoke *stabilize* function. The proposed scheme is tested over different periodicity intervals and results were obtained to evaluate its robustness in an typical delay tolerant network based simulator.

3 Chord Overlay Protocol

Chord is ring like structured overlay protocol based on consistent hashing and provides efficient lookup service. It inimitably maps the identifier space to the set of nodes. The host or a process identified by an IP address and a port number is called a node. Each node possess a unique identifier [30]. When a node is identified by the smallest identifier greater than ‘A’, this node is known as the successor of ‘A’. Chord is the powerful lookup service as it make use of an additional layer that translates high level names into chord identifier [31]. Chord uses consistent hashing [14] to map nodes in a circular identifier space. All nodes maintains a routing table with *m* entries called finger table to implement the successor function [15]. The information about the other nodes are stored in this system. Each entry in the finger table contains node identifier, network address. The *kth* entry in the finger table of the node *K* is the smallest node *r* that is greater than $r + 2^{(k-1)}$. Node *S* is also termed as order-k successor of node *s*. There are *m* unique entries in the finger table. The finger table is in terms of 'm' identifier intervals corresponding to *m* entries in the table. The order of *k* interval of a node *r* is defined as $r + 2^{k-1}$ mod *m*, $r + 2^k$ mod 2^m [8].

The present structure of chord overlay protocol eludes the following problems:

Load Balance The chord overlay protocol spreads the keys evenly over the nodes which provide a natural degree of load balance and so, the chord acts as a distributed hash function.

Decentralization Chord functions as a completely distributed system. No node in the network enjoys priority over the other nodes. Each node has its importance in the network. This perks up the robustness and makes the chord appropriate for loosely organized peer-to-peer applications.

Scalability No parameter tuning is required to achieve scaling as the chord lookup grows as the *log* of number of nodes. Thus the above enables the systems having large number of nodes are also scalable.

Availability The chord protocol is lithe enough to accommodate modifications in the finger table caused by joining of new nodes or failure of existing nodes and sometimes due to major failure in the underlying network. The node responsible for key is found at any worst circumstances. This protocol operates even in the vibrant scenarios.

Flexible Naming There is no constraint on the structure of the keys it looks up the chord key-space is flat and provides large amount of flexibility on how they map their own names to the chord keys.

The chord software is linked with client and server application that uses it. This application intermingle with the chord in two ways. Firstly, it provides the lookup algorithm that returns IP address of the node pertaining to the respective key. Secondly, includes intimation of change in the set of keys for a respective node by the chord software [11]. The mirroring systems balance the load across all the systems, replicate, cache the data and ensure authenticity. Such a system should be fully decentralized to accomplish reliability as there is lack of central authority [11].

3.1 Evaluation of Successor Node

Each node maintains *m* entries in the finger table where $N = 2^m$ and N is the number of identifiers. This represents a significant hashing which requires each node to track almost all the other nodes. Each node contains only the subset of the nodes, evaluating the successor function requires the communication between the nodes at each step of the protocol. The search for the node moves progressively closer to identify it at each step [32]. A search for the node 'f' at node 'r' begins by determining whether 'f' is the immediate successor of 'r'. If So, The search is terminated. Otherwise, 'r' forwards the search request to the largest node in the finger table that precedes 'f'. This search procedure is repeated by 'r' until the search terminates. This algorithm is implemented iteratively. In each iterative implementation, the initiating node is responsible for making request for finger table at each stage of the protocol using *stabilize* function. Iterative approach is easier to implement and relies less on intermediate nodes while the recursive approach leans more towards the server selection and caching. When a new node joins the finger table, it must initiate its finger table and also the other existing nodes must update their finger table to reflect the existence of 'r'. Now, If the system is in a stable state, a new node 'r' uses *stabilize* function and initializes its finger table by querying an existing node for the respective successors of the lower end-points of the 'k' intervals in the *r's* table [33].

3.2 Distributed Hash Service

Chord is not a storage system, it defines the relationship between the nodes and the keys rather than with values. For the fruitful and competent implementation of this relationship between the nodes and the keys a distributed hash table (DHASH) is used. An entry can be added to DHASH using DHASH::insert by hashing the NodeId to produce a 160 bit chord identifier 'k', and storing the value at the successor of 'k'. A DHASH::lookup request is handled analogously and key is hashed to form 'k' and the successor of 'k' is queried for the value associated with the key [33]. An additional RPC interface is used for the transfer of value to data from nodes which is separate from that exported by chord. When the nodes join or leave the system, the successor node of a given key may change. DHASH tracks the joining and leaving of the nodes. Using the call back interface provided by the chord to maintain the in-variant whose values are stored at the successor of their associated keys. Each time when the node joins or leaves the system, An additional cost of transferring O(1/N) keys in the system is imposed by the DHASH layer [34].

4 Chord with Creido Bundles

In the traditional Chord Overlay Protocol, the chances of the bundle to reach the destination is very minimal because it is abstruse whether the bundle from source will reach destined node or not. In this article, a periodic bundle called CREIDO is assimilated. This is transferred between neighboring nodes, updating about the changes in the topology such as new node joining the network and existing node leaving the network. This bundle is kept small in size so as to reduce the bundle traffic and ensure that it does not create hindrance to message bundle communication among the nodes. This light weighted periodic CREIDO bundle is employed because the more weighted finger table cannot be flooded continuously as this will comparatively increase the bundle traffic and leads to congestion in the network.

To make the architecture robust, chord overlay protocol uses *stabilize* function periodically to initiate and update finger table of the new and existing nodes respectively. Let us assume that a new node joins immediately after the run of *stabilize* function. As *stabilize* runs periodically, new node must wait until next run of *stabilize* to initiate its finger table and to notify its neighbor. This new node cannot communicate until the next run of *stabilize*. Similarly, a node departing from the overlay is not detected until the next run of *stabilize*. This adds up to the decline in bundle delivery ratio. The proposed CREIDO bundle is generated by every node and sent to its successors and predecessors periodically. If a node joins, a new CREIDO bundle will be sent to its neighbors invoking the *stabilize* function. So, this new node can instantly initiate its finger table and notifies the neighbors. If an existing node departs, the neighbors will stop receiving the CREIDO thus inferences the departure of node. *Stabilize* function of chord overlay protocol is invoked again to reconstruct the finger table. This periodicity parameter cannot be constant as the periodic interval has to be adjusted according to the network scenario and environment as the same parameter will not produce same results on every scenario. Also, it cannot be made as constant parameter since DTN architecture faces many more challenges as mentioned above in this article. The acknowledgements cannot be considered as the measure of reliability in the DTN architecture as it has the characteristics of high and long disruptions and delays respectively. So, the proposed technique is employed to improve bundle delivery ratio.

The Bundle specification is as follows (see Fig. 1).

VERSION—Indicates the version of the protocol and notifies which version of IP addressing is used.

When set to 0 indicates IP Version 4. When set to 1 indicates IP version 6.

FLAGS—This field indicates the change in topology.

When set to 0 no change in the topology.

When set to 1 change in topology.

SOURCE ID—This field holds the source chord ID.

SOURCE IP—This field holds the hashed destination IP.

DEST. ID—This field holds the destination chord ID.

DEST. IP—This field holds the destination IP address.

5 Simulation Setup

To simulate the proposed scheme of this paper, OverSim [35] a framework for OMNeT++ [36] a generic architecture, object oriented modular discrete event network simulator is used.

5.1 Network Model

In this simulation model simulated area is a free space with nodes moving randomly without a predefined path. Initially, the nodes are placed in such a way that all nodes are connected in the network. Mobile nodes are connected through wireless links and said to be in contact, if they fall within the specified transmission range of each other.

A contact of two nodes holds the following attributes,

Sender The node that contains messages that are to be routed to other node.

Receiver The node which is destined to receive the messages transmitted from sender node.

Time The time at which the two nodes appear in the communication range of each other and start exchanging the control packets.

Duration The time period during which the two nodes are in the communication range of each other and are able to transfer messages. Each node has its own sufficient buffer to store messages. Two attributes of a node's buffer is Capacity and occupancy as defined below.

Capacity The maximum number of messages the buffer of node n can carry.

Occupancy The number of messages in the buffer of node *n* at the beginning of the contact.

Each message *m* has three attributes namely Source, Destination and TTL.

Source The node that generated the message.

Destination The node to which the message should be delivered.

TTL Time To Live field is only used for drop the looping packets.

5.2 Performance Metrics

The metrics considered for the performance analysis of the proposed protocol are bundle delivery ratio, average latency and Throughput, Control Bundle Overhead.

Bundle Delivery Ratio It is the ratio of number of data bundles delivered to the destined node to the number of data bundles generated at the source node during the simulation time

$$BDR = \frac{\sum_{n \in N} P\,(dv)}{\sum_{n \in N} P\,(gn)}$$

Here, N is the set of nodes in the network, n is the node that belongs to *N*, $P(dv)$ is the packet delivered at the destined node, $P(gn)$ is the packet generated at the source node [37].

Throughput It is the fraction of channel capacity used in the total available channel. Mathematically it is the ratio of amount of channel used to the total available amount of channel.

$$AL = \frac{\sum_{n \in N} \sum_{p \in D(n)} C'}{C}$$

Control Bundle Overhead Control bundle overhead is the overhead which occurs due to the CREIDO bundle transits that occurs when the network stabilizes after topology change.

5.3 Mobility Model

An important ramification of any wireless simulation is the mobility model. It becomes an important because mobility increases the capacity of Ad hoc networks [38]. In this simulation three mobility models are used to study the performance of the proposed scheme under different mobility caveats as mobility model plays an important role in bundle delivery [37]. They are as follows.

Random Walk Mobility This Model was first construed mathematically by Einstein in 1926 [35]. This Mobility Model was developed to mimic this erratic movement [39]. In this model, a node moves from its extant location to a new location by randomly choosing a direction and speed of travel. Each node moves in the Random Walk Mobility Model occurs in either a constant time interval *t* or a constant distance traveled *d*. If a node which moves according to this model reaches a simulation boundary, it rebounds off the simulation area border with predetermined angle determined by the incoming direction and continues along a new path [39].

Boundless Area Simulation Model This Mobility Model differs in handling the simulation area. In all other models nodes bounces off or stop moving if they reach simulation boundary. In the Boundless Simulation Area Mobility Model, mobile nodes that reach one side of the simulation area continue traveling and reappear on the opposite side of the simulation area [40].

Nomadic Community Mobility Model This model represents group of nodes that collectively move from one point to another [40]. Here, nomadic nodes roam around a given reference point choosing an entity model [41, 42]. If this reference point changes to a new location, all nomadic nodes in the group travel to a new location defined by the reference point and then starts roaming around the new reference point. The parameters for this model define how far a node roam from the reference point [43].

5.4 Simulation Parameters

In this simulation performance analysis of CREIDO chord protocol over conventional chord protocol is studied by varying the periodicity of CREIDO bundles for different mobility models with performance metrics as described in Sect. 5.2. Table 1 (see Table 1) describes the parameters for simulation model with periodicity of CREIDO set as 10 s,15 s and 20 s with different mobility model mentioned in Sect. 5.3.

Table 1 Simulation Parameters

Parameter	Value		
Examined protocol	CREIDO embedded Chord Overlay Protocol	CREIDO embedded Chord Overlay Protocol	CREIDO embedded Chord Overlay Protocol
Transmission range	150 m	150 m	150 m
Message Bundle size	512 bytes	512 bytes	512 bytes
CREIDO periodicity	20 s	15 s	10 s
No. Of Nodes	25 nodes	25 nodes	25 nodes
Area	5000 m * 5000 m	5000 m * 5000 m	5000 m * 5000 m
Simulation time	1500 s	1500 s	1500 s
Propogation Model	Free space	Free space	Free space
Movement Model	Random Waypoint model, Boundless area model & Nomadic community model	Random Waypoint model, Boundless area model & Nomadic community model	Random Waypoint model, Boundless area model & Nomadic community mode
Pause time	5 s	10 s	30 s
Maximum speed	10 m/s	15 m/s	20 m/s
Reference range for nomadic nodes	500 m	500 m	500 m

6 Performance Analysis and Results

This simulation considers a sparse network with 25 nodes with discrete wireless links. They communicate once they come under transmission range of each other. The impact of packet delivery ratio of the proposed protocol while varying the CREIDO periodicity is studied with different parameters as described in Sect. 5.4 with performance metrics packet delivery ratio, Control Bundle overhead and throughput as defined in Sect. 5.2.

For nomadic community movement model, Random waypoint movement model is used as the entity model. The nodes roam around the reference point up to 500 m.

Figure 2 depicts throughput while the CREIDO periodicity is set as 10 s. In RWP model throughput is increased to 15 % in average. 16 % of average increase in BAS model and 12 % of average increase in NCM model are achieved. Figure 3 depicts the Bundle delivery ratio while the CREIDO periodicity is set as 10 s. While Random waypoint (RWP) mobility model shows an average 23 % of increase in delivery ratio, Boundless area simulation (BAS) and Nomadic community (NCM) mobility models shows 19 and 18 % of average increase in delivery ratio. Figure 4 depicts Control Bundle overhead when CREIDO bundles are transmitted at a period of 10 s. Control overhead rate is averagely increased to 12, 13, 15 % in RWP, BAS, NCM respectively. Even though increase in Control Bundle overhead is a havoc, it is tolerable because average Control Bundle overhead is 25 % in an average. Throughput obtained for the CREIDO periodicity set at 15 s are illustrated in Fig. 5. RWP model has 13 % increased average where as BAS model pulls off 10 % increase in average and NCM model accomplishes 12 % of average improvement. The throughput decreases slightly when the transitions of CREIDO bundle dwindles due to comparative increase in CREIDO periodicity. Bundle delivery ratio while the CREIDO periodicity is set as 15 s is depicted in Fig. 6. While Random waypoint (RWP) mobility model delivers a 17 % increase in delivery ratio, Boundless area simulation (BAS) and Nomadic community (NCM) mobility models attains 19 and 14 % of average increase in delivery ratio respectively.

Figure 7 depicts control packet overhead when CREIDO bundles are transmitted at a period of 15 s. Control overhead rate is averagely elevated to 11, 11.9, 12.3 % in RWP, BAS, NCM respectively. Control Bundle overhead decreases significantly as the CREIDO periodicity increases. Figure 8 depicts throughput while the CREIDO periodicity is set as 20 s. In RWP model throughput is increased to 12 % in average. 9.90 % of average

Fig. 2 Comparison of Throughput—10 s

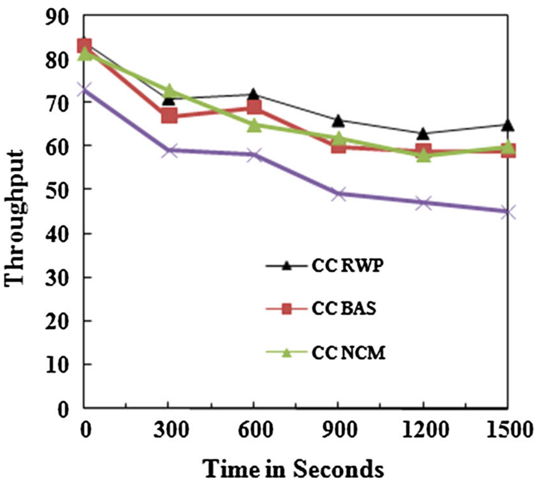


Fig. 3 Comparison of bundle delivery ratio—10 s

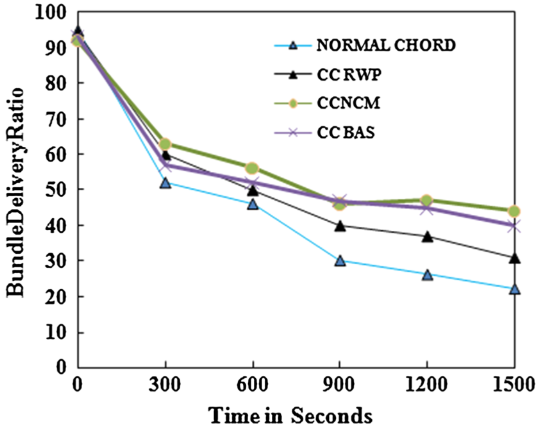


Fig. 4 Comparison of control bundle overhead—10 s

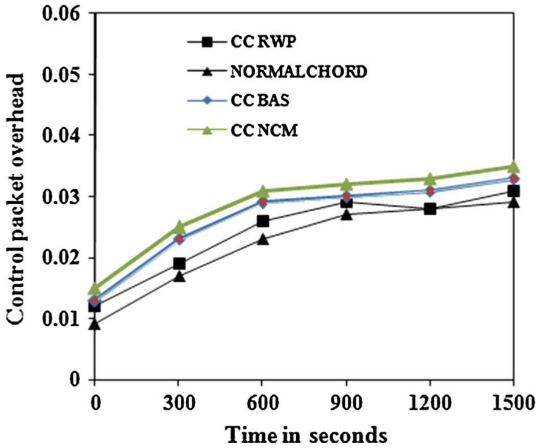
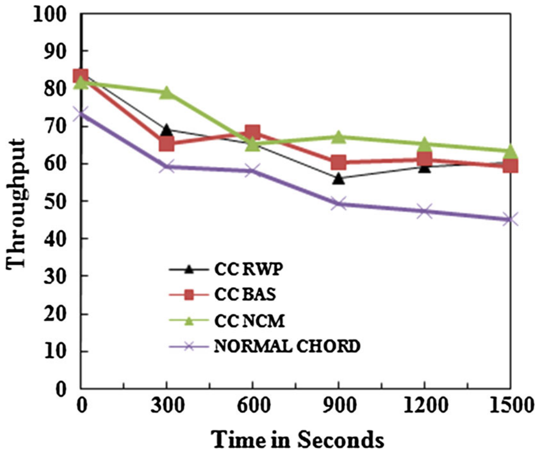


Fig. 5 Comparison of throughput—15 s



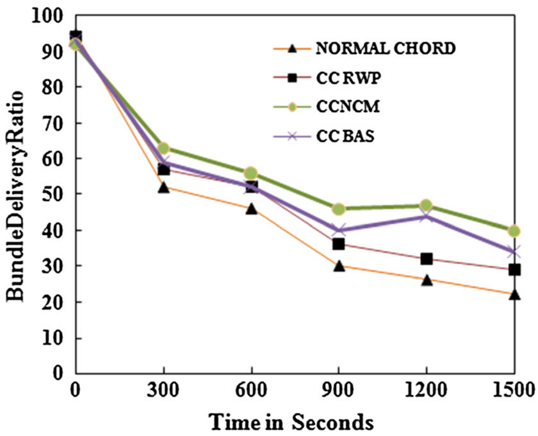


Fig. 6 Comparison of bundle deli- very ratio—15 s

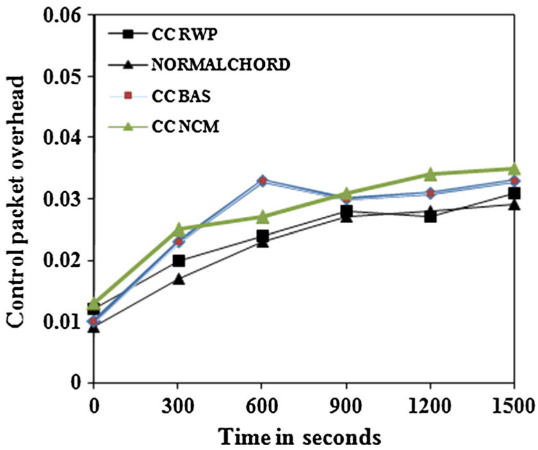


Fig. 7 Comparison of control bundle overhead—15 s

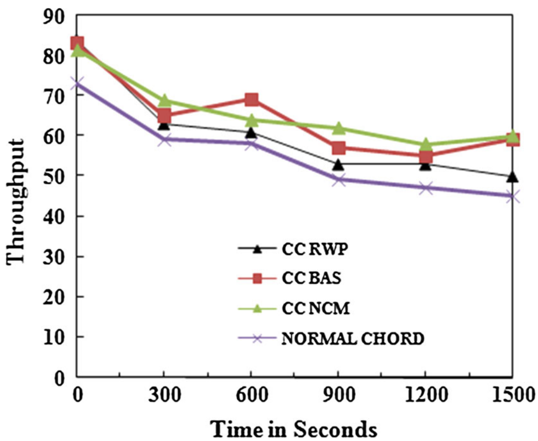


Fig. 8 Comparison of throughput—20 s

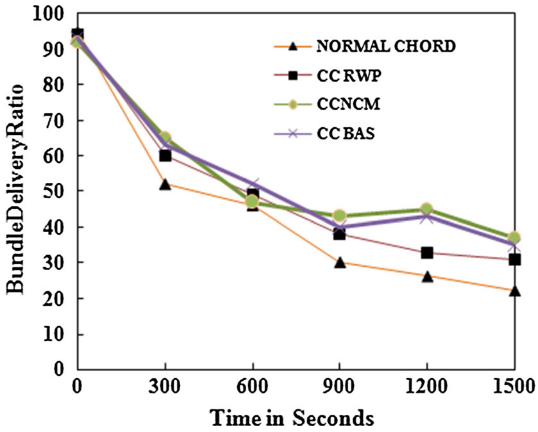


Fig. 9 Comparison of bundle delivery ratio—20 ms

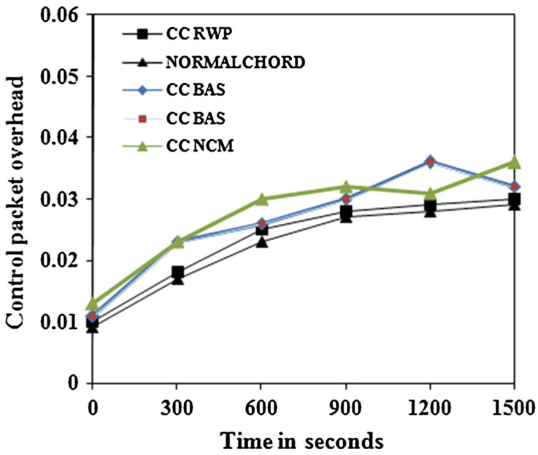


Fig. 10 Comparison of control bundle overhead 20 s

increase in BAS model and 10 % of average increase in NCM model. The throughput slightly decreases as the number of transitions of CREIDO bundle is decreased due to comparative increase in CREIDO periodicity. Figure 9 depicts the Bundle delivery ratio while the CREIDO periodicity is set as 20 s. While Random waypoint (RWP) mobility model shows an average 14 % of increase in delivery ratio, Boundless area simulation (BAS) and Nomadic community (NCM) mobility models shows 12 and 11.5 % of average increase in delivery ratio. Figure 10 depicts control bundle overhead when CREIDO bundles are transited at a period of 20 s. Control overhead rate is averagely increased to 7, 9.97, 12 % in RWP, BAS, NCM respectively. Control Bundle overhead decreases comparatively because of the increase in CREIDO periodicity.

7 Conclusion

In this article periodic CREIDO bundles were introduced over the standard chord overlay protocol to improve the bundle delivery ratio in DTN. The extensive simulations were carried out with the aid of Oversim [21], a framework for overlay simulations on

Omnet++ [22] and quiet interesting results were obtained. The CREIDO based routing outscores the standard protocol when exposed to different types of movement models with reference to the metrics such as Bundle delivery ratio and throughput. An average of 15 % improvement has been accomplished in Bundle delivery ratio and throughput. The future expansion will be focussed on how does the CREIDO based scheme mitigates different attack scenarios which may rise in Delay tolerant networks.

References

1. Raj, V. S., & Chezhian, R. M. (2013). Delay-disruption tolerant networks, its network characteristics and core applications. *International Journal of Computer Science and Mobile Computing, 2*(9), 256–262.
2. Xian, Y., Huang, C. T., & Cobb, J. (October 2010). Look-ahead routing and message scheduling in delay-tolerant networks. In *IEEE 35th conference local computer networks (LCN)*, pp. 40–47.
3. Zhang, Z. (2006). Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges. *IEEE Communications Surveys and Tutorials, 8*(1), 24–37.
4. Pelusi, L., Passarella, A., & Conti, M. (2006). Opportunistic networking: Data forwarding in disconnected mobile ad hoc networks. *IEEE Communications Magazine, 44*(11), 134–141.
5. Nichols, K., Lee Pitts, R., Gifford, K. K., Jenkins, A., & Kuzminsky, S. (April 2010). DTN implementation and utilization options on the international space station. In *SpaceOps conference*, Alabama, 25–30 Apr 2010. doi: 10.2514/6.2010-2173.
6. Fall, Kevin, & Farrell, Stephen. (2008). DTN: An architectural retrospective. *IEEE Journal on Selected Areas in Communication, 26*(5), 826–836.
7. Lua, E. K., Crowcroft, J., Pias, M., Sharma, R., & Lim, S. (2005). A survey and comparison of peer-to-peer overlay network and schemes. *IEEE Communications Surveys and Tutorials, 7*(2), 72–93.
8. Bernhard, I., & Heep, B. (June 2010). dCBR: A global view on network coordinates for more efficient peer-to-peer systems. In *International conference on ubiquitous and future networks (ICUFN)*, Jeju Island, South Korea, pp. 372–377.
9. Klemm, F., Ecole polytech Fed. de Lausanne, Girdzijauskas, S., Le Boudec, J. Y., & Aberer, K. (September 2007). On routing in distributed hash tables. In *Seventh IEEE international conference peer-to-peer computing*, Ireland, 2–5 Sep 2007, pp. 113–122.
10. Akyildiz, I., Su, Y. S. W., & Cayirci, E. (2002). Wireless sensor networks: A survey. *Computer Networks, 38*, 393–422.
11. Mayer, C. P. (2011). *Hybrid routing in delay tolerant networks*. Karlsruhe: KIT Scientific Publishing.
12. Maenpaa, E., & Camarillo, G. (May 2009). Study on maintenance operations in a chord-based peer-to-peer session initiation protocol overlay network. In *IEEE international symposium on parallel & distributed processing*, Rome, 23–29 May 2009, pp. 1–9.
13. Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., Balakrishnan, H. (August 2001). Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of ACM conference on applications, technologies, architectures, and protocols for computer communications (sig- comm)*, San Diego, CA, pp. 149–160.
14. Zhang, S., Wang, B., Wei, G., & Zhou, Y. (October 2011). AF-Chord: An improved Chord model based adjusted finger table. In *4th IEEE international conference on broadband network and multimedia technology (IC-BNMT)*, Shenzhen, 28–30 Oct 2011, pp. 335–338.
15. Dabek, F., Brunskill, E., Kaashoek, M. F., Karger, D., Morris, R., Stoica, I., & Balakrishnan, H. (May 2001). Building peer-to-peer systems with chord, a distributed lookup service. In *Proceedings of the eighth workshop on hot topics in operating systems*, 20–22 May 2001, pp. 81–86.
16. Ktari, S., Zoubert, M., Hecker, A., & Labiod, H. (2007). Performance evaluation of replication strategies in DHTs under churn. In *Proceedings of the 6th international conference on mobile and ubiquitous multimedia*, ACM, pp. 90–97.
17. Knežević, P., Wombacher, A., & Risse, T. (2009). DHT-based self-adapting replication protocol for achieving high data availability. In E. Damiani, K. Yetongnon, R. Chbeir & A. Dipanda (Eds.), *Advanced internet based systems and applications* (pp. 201–210). Berlin: Springer.
18. Chen, D., Tan, Z., Chang, G., & Wang, X. (2009). An improvement to the Chord-based P2P routing algorithm. In *Semantics, knowledge and grid, 2009. SKG 2009. Fifth international conference on*, IEEE, pp. 266–269.

19. Chunling, C., Xu, Y., & Xiaolong, X. (2010). Advanced chord routing algorithm based on redundant information replaced and objective resource table. In *Computer science and information technology (ICCSIT), 2010 3rd IEEE international conference on*, IEEE, Vol. 6, pp. 247–250.
20. Yan, H., Jiang, Y., & Zhou, X. (2008). A bidirectional chord system based on Base-k finger table. In *Computer science and computational technology, 2008. ISCSCT'08. international symposium on,* IEEE, Vol. 1, pp. 384–388.
21. Wu, D., Tian, Y., & Ng, K. W. (2007). An analytical study on optimizing the lookup performance of distributed hash table systems under churn. *Concurrency and Computation: Practice and Experience, 19*(4), 543–569.
22. Dijkstra, E. W. (1982). Self-stabilization in spite of distributed control. In *Selected writings on computing: A personal perspective* (pp. 41–46). New York: Springer.
23. Clouser, T., Nesterenko, M., & Scheideler, C. (2008). Tiara: A self-stabilizing deterministic skip list. In S. Kulkarni & A. Schiper (Eds.), *Stabilization, safety, and security of distributed systems* (pp. 124–140). Berlin: Springer.
24. Alima, L. O., Haridi, S., Ghodsi, A., El-Ansary, S., & Brand, P. (2004). Position paper: Self-properties in distributed k-ary structured overlay networks. In *Proceedings of SELF-STAR: International workshop on self-properties in complex information systems*, Vol. 3460.
25. Janson, T., Mahlmann, P., & Schindelhauer, C. (2010). A self-stabilizing locality-aware peer-to-peer network combining random networks, search trees, and dhts. In *Parallel and distributed systems (ICPADS), 2010 IEEE 16th international conference on*, IEEE, pp. 123–130.
26. Onus, M., Richa, A. W., & Scheideler, C. (2007). Linearization: Locally self-stabilizing sorting in graphs. *ALENEX, 7*, 99–108.
27. Richa, A., Scheideler, C., & Stevens, P. (2011). Self-stabilizing de Bruijn networks. In X. Défago, F. Petit & V. Villain (Eds.), *Stabilization, safety, and security of distributed systems* (pp. 416–430). Berlin: Springer.
28. Kniesburges, S., Koutsopoulos, A., & Scheideler C. (2011). Re-chord: A self-stabilizing chord overlay network. In *Proceedings of the twenty-third annual ACM symposium on parallelism in algorithms and architectures*, ACM, pp. 235–244.
29. Hong, F., Li, M., Lu, X., Yu, J., Wang, Y., & Li, Y. (2004). HP-Chord: A peer-to-peer overlay to achieve better routing efficiency by exploiting heterogeneity and proximity. In H. Jin, Y. Pan, N. Xiao & J. Sun (Eds.), *Grid and cooperative computing-GCC 2004* (pp. 626–633). Berlin: Springer.
30. Liben-Nowell, D. Balakrishnan, H. & Karger, D. R. (July 2002). Analysis of the evolution of peer-to-peer systems. In *Proceedings of the twenty-first annual symposium on principles of distributed computing*, Monterey, CA, pp. 233–242.
31. Zhuge, H., Chen, X., Sun, X., & Yao, E. & HRing, E. (February 2008). A structured P2P overlay based on harmonic series. *IEEE Transactions on Parallel and Distributed Systems*, *19*(2), 145–158.
32. Adam, C., & Stadler, R. (April 2004). Patterns and routing for self stabilization. In *9th IEEE/IFIP network operations and management symposium*, pp. 61–74.
33. Balasubramanian, A. Levine, B. N., & Venkataramani, A. (April 2010). Replication routing in DTNs: A resource allocation approach. *IEEE/ACM Transactions on Networking*, *18*(2), 596–609.
34. Karger, D., Lehman, E., Leighton, F., Levine, M., Lewin, D., & Panigrahy, R. (May 1997). Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web. In *29th annual ACM symposium on theory of computing,* El Paso, TX, USA, May 4–6 1997, pp. 654–663.
35. Baumgart, I., Heep, B., & Krause, S. (May 2007). OverSim: A flexible overlay network simulation framework. In *Proceedings of 10th IEEE global internet symposium (GI '07) in conjunction with IEEE INFOCOM 2007*, USA, pp. 79–84.
36. Varga, A., Hornig, R. (2008). An overview of the OMNeT ++ simulation environment. In *Proceedings of the 1st international conference on simulation tools and techniques for communications, networks and systems & workshops*, Belgium, 2008, pp. 1–10.
37. Abdel-Kader, T. *Design and performance analysis of opportunistic routing protocols for delay tolerant networks*. https://uwspace.uwaterloo.ca/bitstream/handle/10012/6864/Abdel-Kader_tamer.pdf.
38. Grossglauser, M., & Tse, D. N. C. (2002 August). Mobility increases the capacity of ad hoc wireless networks. In *IEEE/ACM transactions on networking*, Vol. 10, No. 4.
39. Sanchez, M., & Manzoni, P. (2001). Anejos: A java based simulator for ad-hoc networks. *Future Generation Computer Systems, 17*(5), 573–583.
40. Davies, V. (2000). Evaluating mobility models within an ad hoc network. *Master's thesis, Colorado School of Mines.*
41. Bai, F., & Helmy, A. *A survey of mobility models in wireless ad hoc networks*. http://www.cise.ufl.edu/~helmy/papers/Survey-Mobility-Chapter-1.pdf.
42. Camp, T., Boleng, J., Davies, V. (2002 August). A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing*, *2*(5), 483–502.

43. Haas, Z. (1997 October). A new routing protocol for reconfigurable wireless networks. In *Proceedings of the IEEE international conference on universal personal communications (ICUPC)*, pp. 562–565.

M. Umaparvathi completed her B.E. (Electronics and Communication Engineering) in the year 1995, M.Tech (Communication Systems) in NIT, Trichy in the year 2005 and Ph.D from Anna University in the year 2013. She has almost 17 years of teaching experience and 2 years of Industrial Experience as Quality Control Engineer. She has attended various workshops and Faculty Development Programmes in Networking. She has been a Resource Person in National Level Workshops for Networks and Simulations. She has acted as Chair in many International and National Conferences. Her research interests are Wireless networks, Information security and Digital Signal Processing. At present, she is working as a Professor in RVS College of Engineering and Technology, Coimbatore.

N. Bhalaji is serving as an Associate professor in the department of Information technology, SSN College of Engineering and Recognized supervisor of Anna University, Chennai. He graduated both his Bachelor of Engineering and Master of Engineering with first class in Computer Science and Engineering and acquired Ph.D degree from Anna university, Chennai under Faculty of information and communication Engineering in the year 2013 under the supervision of Dr. A. Shanmugam (Former Principal Bannari Amman Institute of Technology) in his research thesis he delt with trust based computing in Adhoc networks and currently his primary research focus is towards providing security and privacy in distributed systems and wireless mobile networks. Enhancement of Security in MANETs: Protocols and algorithms for anonymous communication, cooperative forwarding and Misbehavior detection. Application of Trust over information and communication domains namely Internet of Things, Networks and social computing arena. He has published articles in reputed journals and conferences and also serving as an editorial committee member and reviewer in journals published by IEEE society, Elsevier, Springer, CRL publishers. He has initiated a funded project in association with Dr. Chitra Selvaraj titled “Routing Protocol Customization for IoT for societal Applications” Funded by SSN Trust.

S. Jothi Prasanna completed his Bachelor of Engineering in Computer Science Engineering from Anna University, Chennai. He is currently Pursuing Master of Technology in Computer Science and Engineering from SRM University, Chennai. His area of interest is Wireless Ad-Hoc Networks and Internet-Of-Things. He has published articles in Springer and IEEE conferences.

N. Parthiban completed Bachelor of Engineering and and Master of Engineering in Computer Science and Engineering from Anna University, Chennai. Currently pursuing his doctoral studies in VIT University Chennai. Doing his research in 'Graph Theory' in the field of fixed interconnection networks problem. Some of his papers has been published in SCI indexed journals which includes Theoretical Computer Science (Elsevier), The Computer Journal (Oxford Press), Mathematics in Computer Science (Springer) and Journal of Discrete Algorithms (Elsevier).