International Journal of Computational Research and Development (IJCRD)
Impact Factor: 4.775, ISSN (Online): 2456 - 3137
(www.dvpublication.com) Volume 1, Issue 2, 2016

OPTIMIZATION OF THROW BOX DEPLOYMENT IN DELAY TOLERANT NETWORK

S. Sekar* & Dr. C. Poongodi**

* M.Tech Information and Cyber Warfare, Department of Information Technology, Kongu Engineering College, Perundurai, Tamilnadu

** Assistant Professor, Department of Information Technology, Kongu Engineering College, Perundurai, Tamilnadu

Cite This Article: S. Sekar & Dr. C. Poongodi, "Optimization of Throw Box Deployment in Delay Tolerant Network", International Journal of Computational Research and Development, Volume 1, Issue 2, Page Number 68-75, 2016.

Abstract:

Wireless communications have led to the emergence of wireless sensor networks (WSNs), which consist of a large number of sensing devices each capable of detecting, processing, and transmitting environmental information. Delay Tolerant Network (DTN) are designed to overcome limitations in connectivity due to conditions such as mobility, poor infrastructure, and short range radios. DTNs rely on the inherent mobility in the network to deliver packets around frequent and extended network partitions using a store-carry-and forward paradigm. The main goal of DTN is to provide interoperability between different kinds of networks in wide ranging versions and reliable transmission based on overlay network.Throw boxes are very effective in improving throughput and can also reduce data delivery delay. The improvement in throughput is generally more significant than improvement in delay. Throw boxes are most useful for routing algorithms that use multi-path routing and when nodes follow structured mobility patterns. Throw box deployment incorporates knowledge about contact opportunities performs better than deployment. Additionally, if deployment is customized to existing traffic patterns, the algorithms are more effective than assuming that traffic is equally distributed. In this method, these issues are considered, a set of greedy algorithms is proposed which it can efficiently provide quality of solution for challenging problems. To minimize the delivery delay, and to improve reliability in mobile DTN is by placing additional stationary nodes called throw boxes or greater number of contact opportunities with other nodes .This method also investigates the problem of throw box selection and deployment in order to reduce the running time in DTN.

Key Words: Throw Box Optimization, Delay Tolerant Network & Topology Design

1. Introduction:

Delay Tolerant Networking (DTN) is an approach to computer network architecture that seeks to address the technical issues in heterogeneous networks that may lack continuous network connectivity. Examples of such networks are those operating in mobile or extreme terrestrial environments, or planned networks in space. Many evolving wireless network have characteristics different from the internet such as the instability of the link, long propagation and queuing delays, asymmetric data rate and high link errors. DTNs are designed to provide reliable transmission and interoperable communication between wide ranges of networks. DTNs are designed to overcome limitations in Connectivity due to conditions such as mobility, poor infrastructure, and shorter range radios. DTNs rely on the inherent mobility in the network to deliver packets around frequent and extended network partitions using a store-carry-and forward paradigm. However, missed contact opportunities decrease throughput and increase delay in the network. The usage of throw boxes in mobile DTNs is to create a greater number of contact opportunities, consequently improving the performance of the network .The main goal of DTN is to provide interoperability between different kinds of networks in wide ranging versions and reliable transmission based on overlay network. The implementation of DTNs will be good solution for challenged networks. The intermittent connectivity in DTN results in the lack of instantaneous end-to-end paths, large transmission delay and unstable network topology. Recent advances in DTN routing have overcome limitations in connectivity by relying on intermittent contacts between mobile nodes to deliver packets. However, lack of rich contact opportunities in many DTN applications causes poor delivery ratio and long delay of DTN routing.

2. Motivation of Our Work:

Throw box-assisted DTNs are first proposed by [5] where a joint Throw box deployment and routing optimization problem is studied and a greedy algorithm, which relies on network flow technique to solve multiple linear programming problems, is proposed. However, their study only focuses on the average capacity, i.e., the maximum data rate that can be sent between two nodes in long term. Different from them, we consider the detailed topology evolving over time (not just average contact capacity) and aim to optimize the overall routing reliability in the network. In [6], energy efficiency inside each Throw box for Throw box-assisted DTNs is considered. An energy-efficient architecture is proposed and a real test bed is built over such architecture. An approximate heuristic is given for solving the NP-hard problem at a Throw box of meeting an average power constraint while maximizing the number of bytes forwarded. However, their energy optimization is only performed within each individual Throw box. There are also other studies on analytical models of delay distribution [7], [8] and relay strategies [9], [10] for Throw box-assisted DTNs, which do not consider Throw boxes deployment or selection. Various relay placement problems in static wireless networks have been well-studied, such as the static relay placement [11], [12] or the mobile relay planning [16], [17] in static wireless sensor networks. However, the networks studied in this paper are time-evolving DTNs where all devices are mobile and the network topology evolves over time. In our previous work [13], [14], we have studied topology control (TC) problem for time-evolving DTNs, which aims to build a sparse space-time graph while guaranteeing the connectivity or reliability requirement over time. Even though those studies share the underlying space-time graph model with this paper, they are completely different problems. In TC problem, arbitrary links between two nodes at any time slots can be activated or not. In our Throw box optimization problem, if a Throw box is activated, all it's special and temporal links over any time slots are activated. Therefore, the problems, NP-hardness proofs, and solutions are completely different. (iii) In [6],investigates problem with a given set of duty sensors in the plane and in the upper bound of the transmission range to compute the minimum number of relay sensors. Such that the induced topology is globally connected by all sensors. This problem practically considers the trade-off among performance, lifetime, and cost when designing sensor networks. The problem is modelled by a NP-hard network optimization problem. The problem of computing relay sensors is to maintain global connectivity in WSNs when transmission range of all sensors are restricted. (iv) In [7], authors proposed algorithms to deploy stationary Throw boxes in the network that simultaneously consider routing as well as placement. The placement algorithms uses more limited knowledge about the network structure. It performs an extensive evaluation of algorithms by varying both the underlying routing and mobility models. Due to the poor utilization of network resources, epidemic routing achieves the least improvement when using Throw boxes the improvement in throughput is generally more significant than delay. (v) In [8], investigates the performance of a large dense network with one mobile relay and shows the improvement in network lifetime over a static network. Also, implies that the mobile relay needs to stay only within a two hop radius of the sink. Then constructed a joint mobility and routing algorithm which comes close to the upper bound. However it requires all the nodes in the network to be aware of the location of the mobile node. (vi) In [9], Challenges by introducing a capacity-aware routing protocol that is able to search the shortest path considering the time-varying delay and capacity of the virtual links. Using Markov Chain to model the evolution of the real-time link delay, capacity, and use the Markov Chain helps to derive the forwarding decision and routing policy. To evaluate the capacity-aware routing scheme a network graph with the virtual links are extracted from the contact trace and are used. Then introduced a routing strategy that captures and utilizes the time varying link delay and loading capacity for message dissemination with the help of Throw boxes in DTN. The solutions to improve mobile DTN performance is to place additional stationary nodes, called Throw boxes to create a greater number of contact opportunities Throw boxes are small, battery-powered, and inexpensive devices equipped with wireless interfaces and storage. It relay data between mobile nodes in a store-and-forward way. When two nodes pass by the same location at different time, the Throw box acts as a relay, creating a new contact opportunity. Throw boxes can operate without communication with other Throw boxes.Optimization goal of Throw boxes is to maximize the number of packets forwarded. The design and operation of Throw box-augmented DTNs are as follows, Throw boxes are very effective in improving throughput and can also reduce data delivery delay. The improvement in throughput is generally more significant than improvement in delay. Throw boxes are most useful for routing algorithms that use multi-path routing and when nodes follow structured mobility patterns. Throw box deployment that incorporates knowledge about contact opportunities perform better than deployment that ignores this knowledge. Additionally, if deployment is customized to existing traffic patterns, the algorithms are more effective than assuming that traffic is equally distributed.

3. Authors contribution:

In existing system, due to the limitations of power, connectivity, density of nodes, cost, and maintenance, devices may not be able to form a fully connected network for routing data. These network supports delay insensitive applications such as messaging, file transfer, data dissemination, they enables a communication otherwise there may be no communication. Even though DTNs are highly robust to poor connectivity, their performance is highly dependent between nodes. A larger network with more nodes and a larger number of Throw boxes necessitates the use of algorithms to automatically place the Throw boxes. To investigates such algorithms for adding Throw boxes on a running DTN, Throw boxes are placed into an operational network that creates an opportunity to modify the routing technique to utilize the Throw boxes effectively. However, the addition of the Throw boxes affects the flow of data in the network, which consequently affects the placement of Throw boxes. While placement can be considered to be most effective while considering the routing algorithm simultaneously with the placement of Throw boxes, thus maximizing the overall effectiveness of the Throw boxes after they are deployed. To perform an extensive evaluation of placement algorithm by varying the underlying routing and mobility models. The heterogeneous architecture for wireless sensor networks composed of a few mobile nodes and a large number of simple static nodes. These mobile nodes can either act as mobile relays or mobile sinks. To investigate the performance trade-offs associated with the finite network, the lifetime for different routing algorithms are as follows (i) when the network is all static (ii) when there is one mobile sink and (iii) when there is one mobile relay. Using the mobile node as a sink, it results in the maximum improvement in lifetime. Then the performance of a large dense network with one mobile relay and the improvement in network lifetime over an all static network is upper bounded by some factor. Also, the proof implies that the mobile relay needs to stay only within a two hop radius of the sink. Constructing a joint mobility and the routing algorithm which comes much closer to the upper bound. However this algorithm requires all the nodes in the network to be aware of the location of the mobile node. Then proposed an alternative algorithm, which achieves the same performance, but requires only a limited number of nodes in the network to be aware of the location of the mobile. Finally the performance comparison of the mobile relay and mobile sink shows that there is a densely deployed sensor field of radius hops, requires mobile relays to achieve the same performance as the mobile sink. One way to increase the network lifetime is to redeploy more static nodes in the area near the sink. These additional static nodes serve as reservoirs of energy. Normally these nodes are in the sleep state. The nodes near the sink uses their energy these additional nodes to wake up over the relay tasks. To achieve a lifetime improvement by this approach, increase the density of nodes within two hop radius over the sink. It is not easy to see least additional static nodes required to achieve the same performance as the network with one mobile relay. These static relay nodes have greater energy but the same communication range is as like as static nodes. Therefore, the amount of traffic flowing through the original bottleneck nodes will be reduced and the new set of bottleneck nodes, will be the set of neighbors to the sink and also to the set of neighbors of the resource with rich static nodes.

A. Models and Assumptions: In Throw box-assisted DTNs, a joint Throw box deployment and routing optimization problem is studied and a greedy algorithm is proposed, which relies on network flow technique to solve multiple linear programming problems. This method only focuses on the average capacity, i.e., the maximum data rate that can be sent between two nodes in long term. Thus the problem of NP-hardness proofs and their solutions were completely different. This makes routing tasks over them challenging. All Throw boxes have the capacity to buffer any packet for any long time period, thus there exists temporal links of Throw boxes. By defining the space-time graph we also define a reliability probability for each link, which represents the probability of a successful data transmission over link. All Throw boxes have the capacity to buffer any packet for any long time period, thus there exists temporal links of Throw boxes.

- ✓ Throw box Optimization Problems:Active Throw boxes have more forwarding opportunities among devices, by keeping more active Throw boxes usually increases the reliability of the network. The maintenance of active Throw boxes has certain cost in deciding how many active Throw boxes are needed and which Throw boxes should be activated. Through this the Throw box optimization problems over the weighted space-time graph model have defined and have two versions, min-Throw box problem and k-Throw box problem.
- ✓ K-Throw box problem: For k-Throw box problem, where k Throw boxes are to be selected and the goal is to maximize the reliability of the network. The algorithm runs on random networks with n mobile users and m Throw boxes, and k ranges from 1 to 9. For small networks, the optimal solution OPT with brute force algorithm is used. Although they cannot achieve the same level of reliability with these reliability changes, there is a trade-off between network reliability and time complexity. The performance of proposed algorithms in larger in random networks to discover the scalability of the algorithms.
- ✓ Min Throw box problem: For the min-Throw box problem, uses the same sets of random networks, with the reliability constraint g ranging from 0:40 to 0:60. It confirms that more Throw boxes introduces more contact opportunities in the network. The main aim of the OPT is to use smallest number of Throw boxes, however the running time is the largest among other algorithms and increases exponentially. Thus it confirms, using more Throw boxes can introduce more contact opportunities in the network. But the OPT needs the smallest number of Throw boxes to be used.

B. Implementing Greedy Algorithm: By implementing greedy algorithm for the proposed systemare classified as follows: Adding Throw boxes and deleting Throw boxes

- ✓ Adding Throw boxes:With the help of active Throw boxes the forwarding opportunities between nodes and devices increases, thus by keeping more active Throw boxes usually increases the reliability of the network. In order to maintain Throw boxes with certain cost and the requirement of minimum active Throw boxes, the decision have to be taken such as how many active Throw boxes should be needed and which Throw boxes should be activated.
- ✓ Deleting Throw boxes: This algorithm starts with the original space-time graph with all Throw boxes activated, and gradually deletes active Throw boxes until only k active Throw boxes are left or else the reliability constraint breaks in this algorithm. During this process this method greedily deletes one single active Throw box in each round based on certain criteria. These are the operations that are being performed in deleting Throw boxes.

C. Greedy Approach: Finding the optimal solutions in Throw box optimization problem is to explore all the possible combinations of Throw box selection. This is very challenging and time consuming, but the greedy approaches make simple in selecting a single Throw box choice in each round by adding or removing one active Throw box from the network. The procedure will guarantee to terminate after m rounds, which is much more efficient than other exponential brute force algorithm. The same approaches work for both k-Throw box problem and min-Throw box problem, but the only difference is the termination condition. One is when k active Throw boxes are selected, while the other is when the reliability requirement is to be achieved or void.

D. Picking the Best Throw Box: The description of selecting Throw box can be performed into two different criteria in selecting Throw boxes greedily based on their node degrees and their reliability changes. In order to select the best Throw box in each round to be added or to remove from the network the following conditions have to be performed.

- ✓ Based on Node Degrees: Each Throw box may bring new contact and forwarding opportunities to the mobile users in the network. One way to measure such improvement over connectivity of a Throw box is based on its total node degree added to the network using greedy iteration, simply add the Throw box with largest (or remove the Throw box with smallest). The intuition behind the usage of Throw boxes is for better connectivity (larger node degree over time) to improve the reliability among mobile users. The time complexity of Greedy select based on node degrees, the maximum node degree of a Throw box at time t or t+1.
- ✓ Based on Reliability Changes: The reliability changes in adding or removing Throw box in greedy iteration follows adding of Throw box with largest reliability in order to improve the communication time (or remove the Throw box with the smallest deduction). Obviously, this metric is more useful for the optimization goal or constraint with node degrees. For m rounds of n times Dijkstra's algorithm are used. In term of complexity, it much larger than other algorithms based on node degrees. Then it use postfixes to represent greedy criteria using general approach. This greedy algorithm almost uses node degree metric or reliability change metric to select a Throw box that are to be added in each round.

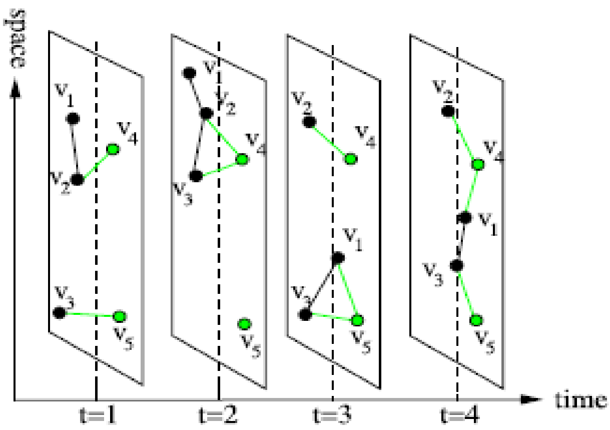

Figure 1: Time-evolving Throw box assisted DTN.

4. Proposed Method:

A. Assumptions: The space-time graph to model the time-evolving DTNs captures the evolving characteristics in both spatial and temporal spaces. Since the positions of individual nodes in the network topology co-evolve over time and the information sequence of static graphs can be defined over several model and their interactions among nodes are calculated in the time evolving DTN. For the reliability of lossywireless links or inaccurate link predictions, defines a reliability probability for each link, which represents the probability of a successful data transmission over link. The reliability probability of each link can be obtained through link estimation techniques at the link layer and physical layers using mobility prediction techniques. With the reliability of each link the reliability of each path or a structure can be established. The reliability for single-copy DTN routing includes only one copy of each message is propagated in the network. Thus, the resulting propagation message path is basically a single space-time path.

- ✓ The reliability of the topology is the minimum path reliability among all source-destination pairs. Another alternative way to define the reliability is taking the summation of path reliabilities instead of the minimum. It is easy to calculate by using any shortest path algorithms.

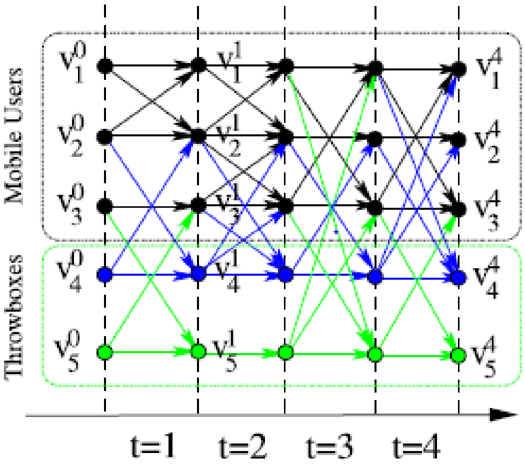

Figure 2: Selecting one active Throw box.

B. General Greedy Approaches: Finding the optimal solutions for Throw box optimization problem by exploring all possible combinations of Throw box selection is very challenging and time consuming, thus, our greedy approaches simply make a single Throw box choice in each round by adding or removing one active Throw box from the network. The procedure will guarantee to terminate after at most m rounds, which is much more efficient than exponential brute force algorithm. The same approaches work for both k-Throw box problem and min-Throw box problem, and the only difference is the termination condition. One is when k active Throw boxes are selected, while the other is when the reliability requirement is achieved or void.

C. Greedy-Adding Throw boxes (GrdAddTBs) Given a weighted space-time graph G (with n mobile users and m potential Throw boxes) and a threshold, the aim of min-Throw box problem is to find the minimum set of active Throw boxes such that the space-time graph H formed by these Throw boxes and all mobile users

D.Greedy-Adding Throw boxes (GrdAddTBs)

***Input*:** The original space-time graph G (including potential Throw box set $V_{\text{Throw box}}$), a constant k (or a threshold g).

Output:The selected Throw box set $V_{\text{selected\_Throw box}}$ and the corresponding new space-time graph H.

1: H ← ~~G~~ – {$V_{\text{Throw box}}$} and {$V_{\text{selected\_Throw box}} = \emptyset$}

2: **while** | $V_{\text{selected\_Throw box}}$ | < k do

3: Greedily select a Throw box V_i from all unselected Throw boxes$V_{\text{Throw box}} - V_{\text{selected_Throw box}}$, i.e., V_i = GreedySelect($V_{\text{Throw box}} - V_{\text{selected\_Throw box}}$; H)

4: H ← H + {V_i}

5: $V_{\text{selected\_Throw box}} V_{\text{selected\_Throw box}}$ + {V_i}

6: **return**$V_{\text{selected\_Throw box}}$ and H

This algorithm (GrdAddTBs) starts with a space-time graph H only including mobile users (v_1,v_2,…,v_n). Then it greedily adds in active Throw boxes until either k Throw boxes are activated (for k-Throw box problem) or the reliability of H reaches the required threshold (for min-Throw box problem).

E. Greedy- Deleting Throw boxes (GrdAddTBs):

Input: the original space-time graph G (including potential Throw box set $V_{\text{Throw box}}$), a constant k (or a threshold g).

Output: the selected Throw box set $V_{\text{selected \_Throw box}}$ and the corresponding new space-time graph H.

1: H← G and $V_{\text{selected \_Throw box}} = V_{\text{Throw box}}$

2: while |$V_{\text{selected \_Throw box}}$ | > k (or r (H) > ℧) do

3: Greedily select a Throw box $v_i$from$V_{\text{selected _Throw box}}$, i.e., v_i = Greedy Select ($V_{\text{selected \_Throw box}}$, H)

4: H← H – {v_i }

5: $V_{\text{selected \_Throw box}}$ $V_{\text{selected \_Throw box}}$ – {v_i }

6: return ← $V_{\text{selected \_Throw box}}$ and H (or $V_{\text{selected \_Throw box}}$ + {v_i} and H {v_i }

The second algorithm (GrdDelTBs) starts with the original space-time graph G with all Throw boxes activated, and gradually deletes active Throw boxes until only k active Throw boxes are left or the reliability constraint breaks. In both algorithms, during the process, our method greedily selects one single active Throw box in each round based on certain criteria. Hereafter, we generalize such greedy selection of a single Throw boxv_i from a set of Throw boxes v_x based on current space-time graph H as a function Greedy Select with $v_i$as its output. Let us denote the time complexity of Greedy Select as X. Both GrdAddTBs and GrdDelTBs can obviously satisfy the number of Throw boxes requirement (or reliability requirement) of H.

F. Simulations: We first test our algorithms on randomly generated networks. We generate a sequence of static random graphs with n + m nodes (n mobile users and m potential Throw boxes) over T = 10 time slots. For each static snapshot, the link between two mobile users or one mobile user and one Throw box is randomly inserted based on a probability p. Clearly, the larger value of p is, the denser the network is. We test different settings of these parameters in our simulations, and the discoveries and conclusions are consistent. Due to space limit, we only report the results for the following setting. We set p = 0:11, r_{min} =0:3, andr_{max} =0:6 for links between a pair of mobile users; and set p = 0:22; r_{min} =0:6; andr_{max} =1:0 for links between a mobile user and a Throw box. Obviously, Throw boxes are usually more reliable than normal mobile devices. Finally, we generate the weighted space-time graph based on the sequence of static graphs. For each setting, we generate 100 random time-evolving networks and report average performances of our proposed algorithms.

GrdAdd(Del)TBs-D: Greedy algorithms adding/deleting Throw boxes based on node degrees.
GrdAdd(Del)TBs-R: Greedy algorithms adding/deleting Throw boxes based on reliability changes.
GrdAdd(Del)TBs-Ra: Greedy algorithms randomly adding/deleting Throw boxes.
GrdAddTBs-P: Greedy algorithm adding Throw boxes based on reliable path with least Throw boxes.
OPT: The optimal solution for Throw box optimization problem
Obtained by brute force method.

5. RESULT ANALYSIS:

A. Energy Consumption: Here n mobile users and their contact patterns are given as the input space-time graph and the selection is only made over Throw boxes.The reliabilities reached by the algorithms GrdAddTBs-Ra, GrdDelTBs-Ra, GrdAddTBs-R, GrdDelTBs-R are shown. Figure 3 shows GrdAddTBs-R and GrdDelTBs-R algorithms can achieve the best reliability when compared to GrdAddTBs-Ra, GrdDelTBs-Ra.As number of Throw boxes increases the reliability rate will also increase but only in a sufficient manner which also consumes maximum running time.

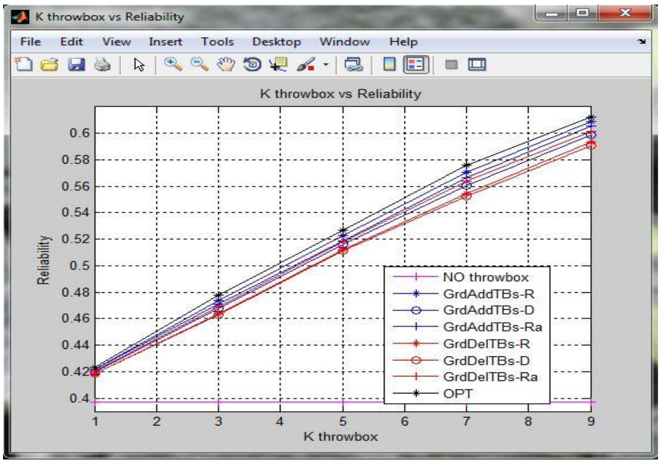

Figure 3: Reliability for different Throw boxes.

B. Reliability for Minimum Throw Boxes: For every timeslots, distance between each node is calculated based on the Euclidean distance between two mobile nodes or one Throw box and one mobile nodes.

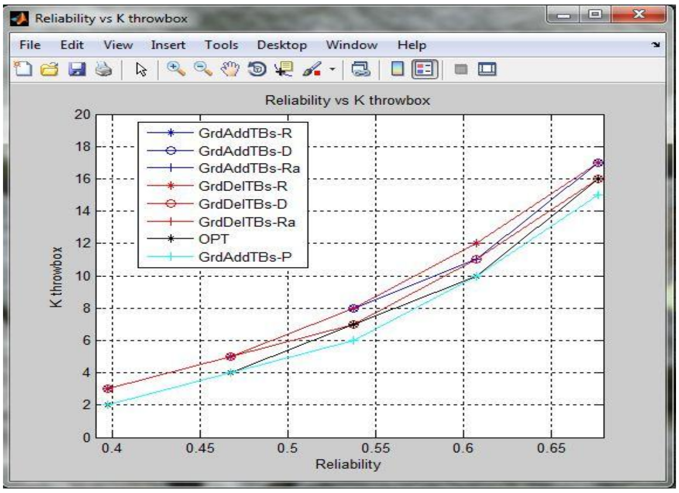

Figure 4: Reliability for minimum Throw boxes

From figure 4, it has been concluded that using minimum Throw boxes the reliability achieved is higher than compared to k-Throw box problem. By using minimum number of Throw boxes the reliability among the routing path between Throw boxes and mobile users will also be increased.

C. Effective Running Time: Finally, by using this proposed Greedy algorithm, the delay tolerant network has been implemented where optimized reliability and running time is assured.

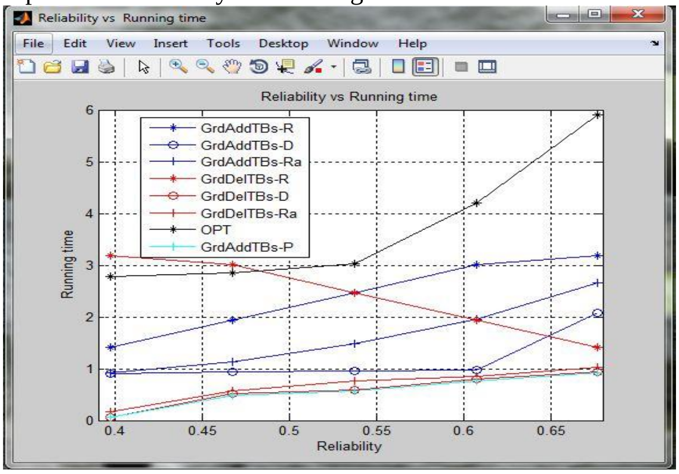

Figure 5: Effective Running Time

Figure 5 shows the running time improved by using minimum Throw boxes which also prolongs the maximum reliability. The performance of the proposed algorithm in larger network to discover the scalability of algorithms. GrdAddTBsR, GrdDelTBs-Rachieves high reliability.

6. Conclusion:

In this approach the investigation of key problem, Throw box selection in a time-evolving Throw box assisted DTN are modeled by a weighted space-time graph. This method formally analyze the hardness of both problems and propose a set of greedy algorithms which can efficiently provide quality solutions. The efficiency of the proposed methods through extensive simulations over random time-evolving DTNs are being carried out and it increases the reliability and throughput.

7. References:

1. Baccour N, Koubaa A, Mottola L, Youssef H,Boano C H and Alves M (2012),"Radio link quality estimation in wireless sensor networks: A survey", ACM Transactions on Sensor Networks,Vol. 8, No. 4, pp.34:134:33.
2. Banerjee N, Corner M D, and Levine B N (2010), "Design and field Experimentation of an energy-efficient architecture for DTN Throw boxes", IEEE/ACM Transactions on Networking, VOL. 18, NO. 2, pp.554–567.
3. Cheng X, Du D Z, Wang L and Xu B (2008), "Relay sensor placement in wireless sensor networks" ,Wireless Networks, Vol.14, No. 3, pp. 347–355.
4. Huang M, Chen S, Zhu Y and Wang Y (2013), "Topology control for time evolving and predictable delay-tolerant networks", IEEE Transactions on Computers, Vol. 62, No.11, pp. 2308–2321.
5. Li F, Chen S, Huang M, Yin Z, Zhang C and Wang Y (2015), "Reliable topology design in time-evolving delay-tolerant networks with unreliable links", IEEE Transactions on Mobile Computers, Vol.14, No. 6, pp. 1301–1314.
6. Lloyd E L and Xue G (2007), "Relay node placement in wireless sensor Networks", IEEE Transactions on Computers, Vol. 56, No. 1, pp. 134–138.
7. Merugu S, Ammar M and Zegura E (2014), "Routing in space and time in networks with predictable mobility", Georgia Institute of Technology, Atlanta, Georgia, USA, GIT-CC-04-07.
8. Wan P J, Du D, Pardalos P and Wu W (2010), "Greedy approximations for Minimum submodular cover with submodular cost", Computer Applications,Vol. 45, No. 2, pp.463–474.
9. Wang W, Srinivasan V and Chua K C (2005), "Using mobile relays to prolong the Life time of wireless sensor networks", 11th Annual International Conference on Mobile Computer Networks, pp. 270–283.
10. Zhao W, Chen Y, Ammar M, Corner M, Levine B and Zegura E (2006), "Capacity Enhancement using Throw boxes in DTNs", IEEE International Conference on Mobile Adhoc Sensor System, pp. 31–40.
11. G. Nemhauser, L. Wolsey, and M. Fisher (1978), "An analysis of approximations for maximizing submodular set functions-I", Mathematical Programming, vol. 14, no. 1, pp. 265–294.
12. L. A. Wolsey (1982), "An analysis of the greedy algorithm for the submodular set covering problem", Combinatorica, vol. 2, no. 4, pp. 385–393.

13. P.J. Wan, D.Z. Du, P. Pardalos, and W. Wu (2010), "Greedy approximations for minimum submodular cover with submodular cost", Computer Applications,vol. 45, no. 2, pp. 463–474.
14. F. Kuipers, P. Van Mieghem, T. Korkmaz, and M. Krunz (2002), "An overview of constraint-based path selection algorithms for QoSrouting", IEEE Communication management, vol. 40, No. 12, pp. 50–55.
15. D. S. Reeves and H. F. Salama (2000), "A distributed algorithm for delay constrained unicast routing", IEEE/ACM Networks, vol. 8, No. 2, pp. 239–250.
16. Vahdat and D. Becker (2000), "Epidemic routing for partially connected ad hoc networks", Duke University, Durham, NC, USA, CS200006.
17. T. Spyropoulos, K. Psounis, and C. S. Raghavendra (2008), "Efficient routing in intermittently connected mobile networks: The multiple-copy case", IEEE Transactions on Networks, vol. 16, no. 1, pp. 77–90.
18. X. Zhang, G. Neglia, J. Kurose, and D. Towsley (2007), "Performance modeling of epidemic routing," Comput. Netw, vol. 51, no. 10, pp. 2867.
19. A. Agrawal and R. E. Barlow (1984), "A survey of network reliability and domination theory," Oper. Res., vol. 32, no. 3, pp. 478–492.
20. Y. Zhu, C. Zhang, X. Mao, and Y. Wang (2015), "Social based Throw box placement schemes for large-scale mobile social delay tolerant networks," Computer Communications, vol. 65, pp. 10–26.