International Journal of Computing and Network Technology

http://dx.doi.org/10.12785/ijcnt/010205

Space Optimized Multicast in Delay Tolerant Networks

Anshuman Tripathi

Department of Computer Science and Engineering, IIT Kharagpur.
Email Address: anshumant.iit@gmail.com

Received: 12 Jan. 2013, Revised: 12 Feb 2013; Accepted: 16 Mar. 2013
Published online: 1 May 2013

Abstract: Many multicast protocols for Delay Tolerant Networks fall in the class of tree based algorithms. Most of these tree computations aim to reduce the time taken by the message to reach its recipients. However, many scenarios have a greater demand to save on the space usage than on the time requirement. The present work proposes a tree computation technique that aims to reduce the duplication of messages in the network as a result of store-and-forward mechanism in tree based multicast. The proposed algorithm has been simulated on SLAW mobility model and shows a reduction in the number of copies of message in the network by as much as 75% (reducing to ¼ th of the number), compared to the Dynamic Tree Based Routing protocol.

Keywords: Space Optimized, Multicast, Delay Tolerant Network, Node importance, Tree-based multicast.

I. Introduction

Delay Tolerant Networks [1] (abbreviated as DTN from here on) are a special class of networks that witness huge churn on availability of links. In such networks, there is no guarantee of an instantaneous path from source to the destination, which makes the problem of routing, broadcasting and multicasting much more complex. The routing protocols use store-and-forward mechanism to transmit messages from source to the destination, thus creating extra copies of messages with each forward. However, since the nodes participating in a DTN are mostly heterogeneous and mobile nodes, with storage constraints, some scenarios of multicasting require minimizing the storage usage on the nodes. Such use cases can arise in situations when either there is a resource constraint on the nodes storage, or the general size of multicast messages is very large. In this paper, a generic tree computation algorithm has been proposed for computation of multicast paths with lower number of intermediate nodes, in tree-based multicast protocols. The paper also presents two specifications of the algorithms and evaluates the efficacy of these modifications on reducing the number of copies of message spawned, and also the extent of compromise that these algorithms make on the message delivery delay, as compared to the Dynamic Tree Based Routing (referred to as DTBR from here on).

II. Realated Work

The problem of multicasting has been studied in detail for DTNs, ad-hoc networks and Internet [2]–[6]. Many multicasting protocols have been developed like epidemic routing [7], Spray-and-wait [8] relay cast routing [9] and direct delivery. More sophisticated algorithms have been developed that specialize in one or many aspects of multicasting. Like, Context Aware multicasting [10] optimizes the delivery ratio based on the mobility patterns. Custodial Multicast [11] ensures that the message reaches the destination. Similarly studies have been done on exploiting weak global knowledge of the DTNs and improving the multicast and routing performances of algorithms. Algorithms like PROPHET [12] requires each node to have a weak knowledge about the probabilities of contacts of various nodes in DTN. Many multicast protocols that utilize this concept of global knowledge, fall under the category of tree based multicast algorithms like Dynamic Tree Based Routing (DTBR), Static Tree Based Routing (STBR), Ondemand Situation-aware Protocol (OSP) [13] and others that compute speculative multicast paths based on some weak knowledge of the global state. Most of these treebased algorithms developed for multicasting use the message delivery time as the criteria for optimization. This approach, however, generally leads to many nodes participating in the multicast event. There can be requirement of minimizing the message copy generation in some space constrained multicast scenarios. In this paper, a novel technique is presented through which delivery time can be compromised to an acceptable level, for achieving a lower number of nodes in the multicast tree.

III. SYSTEM MODELS

3.1. DTN model

In this work the DTN is viewed differently at the physical and application layer. At the physical layer each node can transmit messages to its neighbours. Each node has a Bluetooth-like wireless interface that doesn't allow transmission of data beyond a finite range. It is assumed that within the boundaries of the wireless signal the data transfer rate is constant.

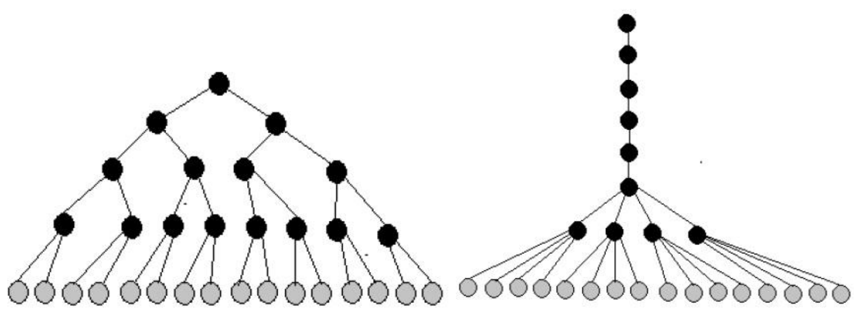


Figure 1. Thin trees have lesser participating nodes

At application layer, the DTN is represented as a complete weighted graph, where the vertices of the graph denote the actual mobile nodes, and the weighted edges denote the average time between consecutive contacts of respective nodes.

3.2. Multicast Model

Multicast refers to one-to-many delivery of a message. It is assumed that there is no global concept of multicast groups. Any source can send message to any sub-set of nodes (which are then depicted in the message header). Each node, therefore, has a unique ID that identifies it in the network.

IV. PROPOSED ALGORITHM

4.1. Motivation

The main motivation of the algorithm is to re-utilize a path that has already been discovered, to potentially more recipients, thereby reusing the copies of messages already spawned. A multicast tree with such a property will generally be ”thinner” close to root and ”wider” near leaves. The algorithm captures the ideas of such a requirement by associating with each node a property referred as **Importance**, which dictates the likelihood that the node will be used in a multicast path. Thus, by suitably modifying the *importance* of a node, desired tree structures can be obtained. As depicted in Figure 1, such tree will have higher latency but lower number of participating nodes, and thus lower number of message spawns.

4.2. Importance

Importance of a node is a measure of the preference for its selection, that shall be given to the node in computation of the multicast tree. Consider the general optimization function for problem of minimizing delivery time:-

$$\textit{minimize}\ W_P = \sum\nolimits_{e(n_1,n_2)\in P} W_e \tag{1}$$

here W_e is the weight function for an edge, P is a path from source to a destination and e is an edge in the path. By introducing the concept of importance, optimization Function 1 can be modified as:-

$$\textit{minimize}\ W_P = \sum\nolimits_{e(n_1,n_2)\in P} W_e / \max(I_{n_1}, I_{n_2}) \tag{2}$$

In this optimization, the effective weight of an edge is lowered, if the importance of its nodes is high. Thus a node which has a high value of importance parameter, is more likely to be selected in the optimization.

4.3. Algorithm Description

In this section, a detailed pseudo-code description of the modified tree computation algorithm is presented in Algorithm 1, referred from here on as SMTBR (Space Minimizing Tree Based Routing). It is assumed that the importances are initialized to 1 for each node in the graph.

Algorithm 1 *Tree computation in SMTBR*

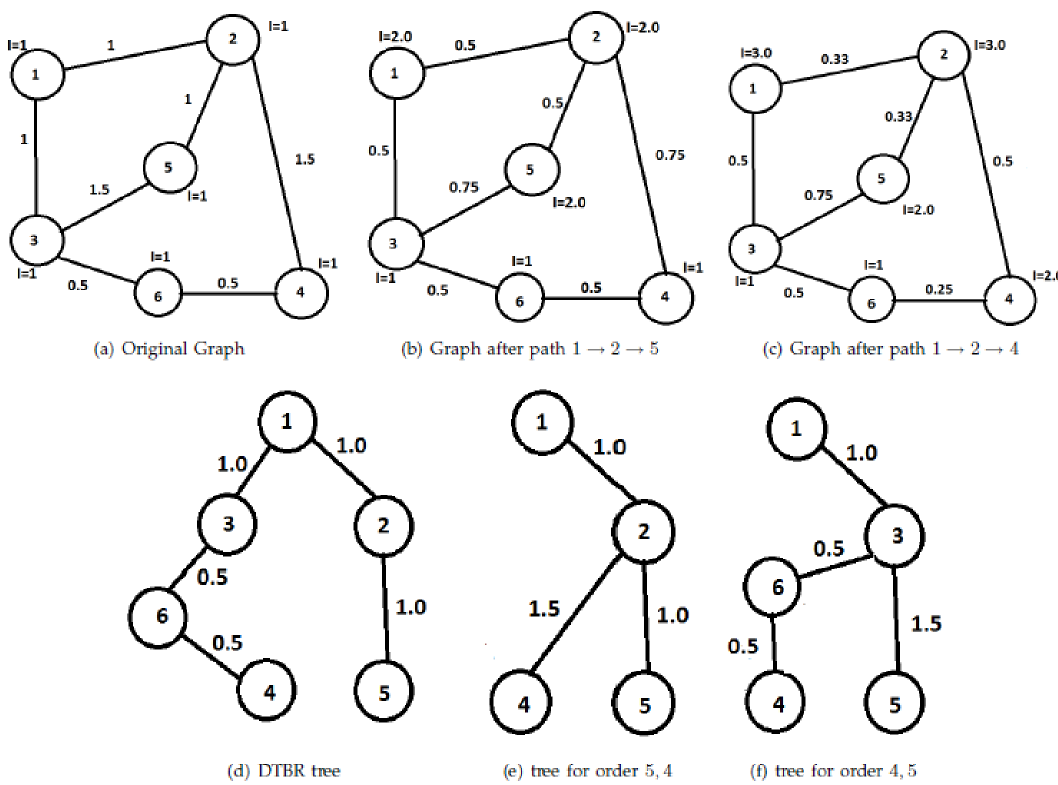
```
{Given: Source s, Destinations d[], Contact Graph G, Importance
Update Function F_I()}
{Initialize the importance of all the nodes to 1.}
G' ← G
for target in d[] do
    path ← shortest path from s to target inG'
    {updating Importances}
    for node in path do
        I[node] ← F_I(I[node])
    {updating graph}
    for edge in E(G') do
        n1, n2 ← Nodes for Edge edge
        G'[edge].weight ← G[edge].weight/max(I[n1], I[n2])
T ← shortest path tree from s to d[] in graph G'
Return T as the tree.
```

4.4. Specifications

Based on the function to update the importance of a node, the following two important specifications of the algorithm are studied in this paper.

4.4.1 SMTBR$_1$

For this specification, the node update function is chosen as $I_{new} = I_{old} + \alpha$, where value of α is fixed for a computation. The main motivation behind this choice of function is that in every iteration, the importance of the nodes that are already in the multicast tree increases as they are selected. This particular update of importance induces a recursive dependence on in the sense that an increase in the importance shall improve the chances that the node is selected again in the next iterations and the selection of node again increases the importance. This shall logically result in formation of hubs (nodes along which most of traffic is routed) thus reducing the number of redundant copies. This, as shall be demonstrated in later sections, is actually the case.



(a) Original Graph (b) Graph after path 1 → 2 → 5 (c) Graph after path 1 → 2 → 4

(d) DTBR tree (e) tree for order 5, 4 (f) tree for order 4, 5

Figure 2. Example: computing SMTBR1(α = 1:0) tree with 1 as source node and 4; 5 as destinations

4.4.2 SMTBR$_2$

Another choice for the importance update function of a node is $I_{new} = \beta$, where the value of β is fixed for a computation. This version of the algorithm is also logical in the sense that it only distinguishes important nodes from unimportant nodes and makes no distinction among the important nodes. When a node is selected in a multicast path, the importance of the node is statically changed to $I_{new} = \beta$, where after it remains constant. This means that nodes in the graph can be partitioned in important nodes $(I = \beta)$ and unimportant nodes $(I = 1)$. Because of the formulation, it is expected that the sensitivity towards β is lesser in this case.

4.5. Example

Figure 2(a) shows an example input graph (all importances are initialized to 1) to the algorithm described. Path computations are shown for the *SMTBR$_1$*(α = 1.0) version. Similar computations can be done, for *SMTBR$_2$*(β = 2.0) specification. The nodes are numbered 1 to 6. The source node is 1 and the target nodes are 4 and 5. If the shortest path algorithm is used, then the tree formed is shown in Figure 2(d), having total of 6 participating nodes. However in *SMTBR$_1$* following steps are followed:

1. Find shortest path route from 1 to first target node (say 5) giving route: 1 → 2 → 5.
2. Update the importance of the nodes in the route (viz. 1, 2 and 5) by adding 1.0 to the old values of importance.
3. Update the edge weights in the graph, by dividing the weight of an edge from maximum of the importances of its endpoint nodes. The resulting graph is given in Figure 2(b). The graph shows the updated values of Importances and the resulting effective edge weights.
4. Now repeat these steps for the destination node 4. The final graph is given in Figure 2(c).
5. Now use the usual shortest tree algorithm to find shortest paths from 1 to 4 and 5 in Figure 2(c). This results in a tree as shown in Figure 2(e).

Note that the sequence in which the nodes are considered in the algorithm, also affects the final tree computed. For example, if the target nodes were considered in the order of: 4 then 5, the tree computed would be as shown in Figure 2(f). Also note that the *SMTBR$_1$* tree has lesser number of nodes (thus lesser message copies are spawned). The number of nodes, however, depend on the order in which target nodes are considered, as evident from Figure 2(e) and 2(f).

Table 1. SLAW trace

Number of nodes	1000
Area of simulation	4000m×4000m
Warmup time	1000s
Simulation Time	259200s (3 days)
Number of waypoints	50
Alpha	3
Hurst parameter	0:75
Clustering Range	50m
Beta (for pause time)	1s
Min wait time	30s
Max wait time	1hr

Tree for the $SMTBR_2$(β = 2.0) variant can also be worked out in this way, only difference being that while updating the Importance values, instead of adding something to the old values, just set the new values as 2.0. In this case also, the trees computed would be same as that for $SMTBR_1$.

V. Simulations And Results

In this section, the investigation of $SMTBR_1$, $SMTBR_2$ and *DTBR* algorithms is presented based on message copies produced and the delivery latency. The simulation results are presented for a SLAW mobility model with parameters as described in Table 1 (generated from [14]). The results presented are averaged over 100 multicast events, each with 100 recipients, with 100% delivery. The experiments are done using a custom designed event based simulator.

From here on the number of message copies, denote the message copies generated in only the store-forward nodes (message storage on recipients and source is not accounted). Delivery latency is the total time that is required for completion of the multicast event.

5.1. Message Copies Spawned

Since the main motive behind the new algorithm was to reduce the number of message copies in multicast events. In this section, results for the performance of the proposed algorithms and the *DTBR* protocol on this aspect are presented. Notice that *DTBR* is just a special case of $SMTBR_1$ (for α = 0) and $SMTBR_2$ (for β = 1).

Figures 3 and 4 depict the plots for number of message copies spawned when the parameter for corresponding algorithm specification is varied. From these plots it can be clearly seen that as the value of (for $SMTBR_1$) and β (for $SMTBR_2$) is increased, the number of messages spawned in the multicast event reduces dramatically in the beginning an then shows a saturation. The decrease in the number of copies provides an evidence for the efficacy of the proposed techniques. Since there is a minimum possible value for the total number of nodes in the multicast tree for a contact graph, the plots show a saturation in the number of message copies as the value for parameters of the algorithms is increased. Notice that the number of message copies can be reduced to as much as 25% of the original (from around 45 to just 11).

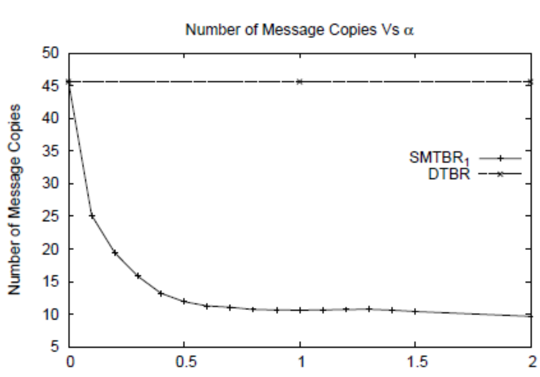


Figure 3. Message Copies for $SMTBR_1$ with α

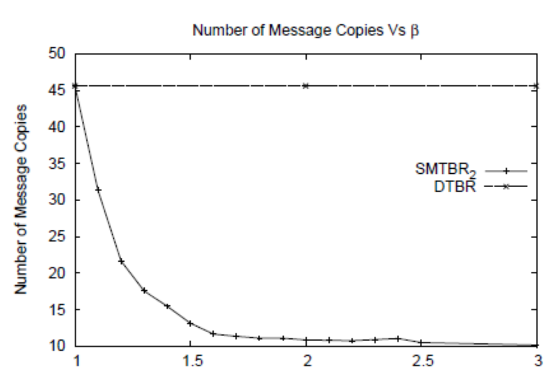


Figure 4. Message Copies for $SMTBR_2$ with β

5.2. *Message Delivery Latency*

As the main motive for the proposed techniques is to reduce the number of message copies, these algorithms compromise on the message delivery latency for the multicast event in the network. Figures 5 and 6 show the plots depicting the increase in the message delivery latency as the corresponding parameters for the $SMTBR_1$ and $SMTBR_2$ are varied.

Figure 7 shows the inter-relationship between the achieved reduction in message copies and the corresponding increase in the message delivery latency, in form of a plot from the data points of both $SMTBR_1$ and $SMTBR_2$. It can be seen from these plots that for both the algorithms, as the number of message copies reduces, the average delivery time increases more rapidly. This shows a ressemblance to the law of *diminishing returns*, that is when the parameter for the algorithm is varied, there is a rapid drop in the message copies in the network in beginning, but as a state of saturation is reached in the number of copies, the average delivery latency still increases. Hence its not very profitable to keep the values of parameters for $SMTBR_1$ and $SMTBR_2$, very high.

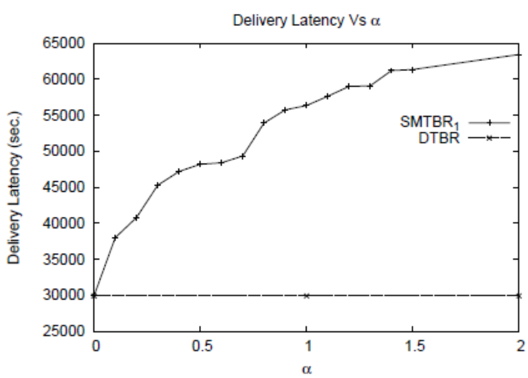


Figure 5. Delivery Latency of $SMTBR_1$ with α

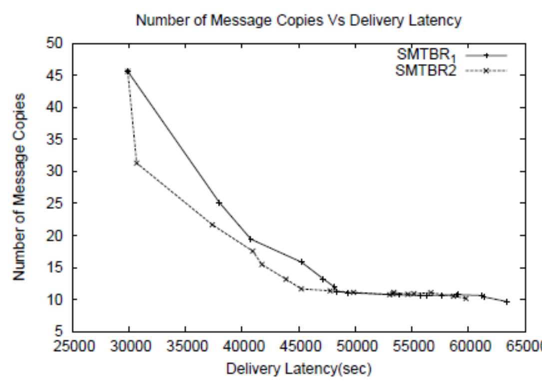


Figure 6. Delivery Latency of $SMTBR_2$ with β

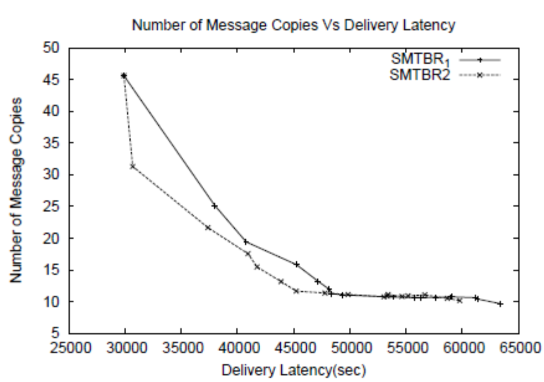


Figure 7. Delivery Latency Vs Message Copies for $SMTBR_1$ and $SMTBR_2$

Figure 7 also shows that in general the $SMTBR_2$ protocol achieves a better performance in reduction of message copies, i.e for the same level of reduction in copies, the latency in $SMTBR_2$ is slightly better than that in $SMTBR_1$. This observation can be reasoned on the fact that, in $SMTBR_1$ the importance of a node increases in every iteration, it is selected in a path, resulting in more deviation from the minimum delivery latency tree. However, in $SMTBR_2$ protocol, the importance of a node is either β or 1, and once it has reached a level of β it doesn't increase.

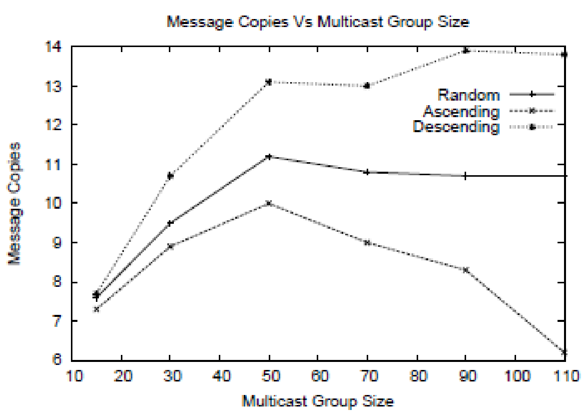
5.3. Effect of Target Ordering

As evident from Algorithm 1 described in section 4.3, it can be well reasoned that the order, in which the paths for destinations are considered (*for* loop at step 4), may have a effect in the performance of the algorithm. All the results described herein above are with random ordering of the target nodes. In this section three types of target orderings for both $SMTBR_1$ and $SMTBR_2$ are discussed:

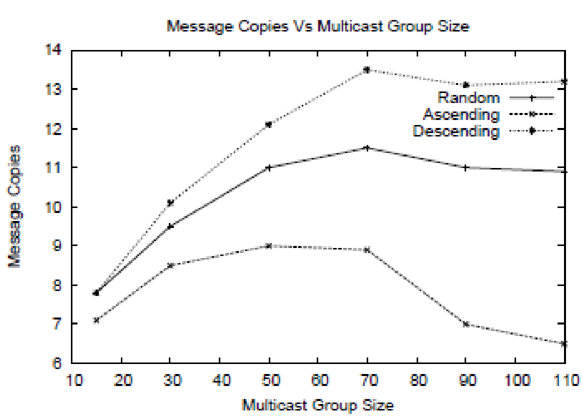
1. Increasing order with respect to the shortest path from the source.
2. Decreasing order with respect to the shortest path from the source.
3. Random order.

5.3.1 Message Copies

In this section, the effect of order of targets is examined on the number of nodes in the computed tree (which corresponds to the number of messages created in that multicast).

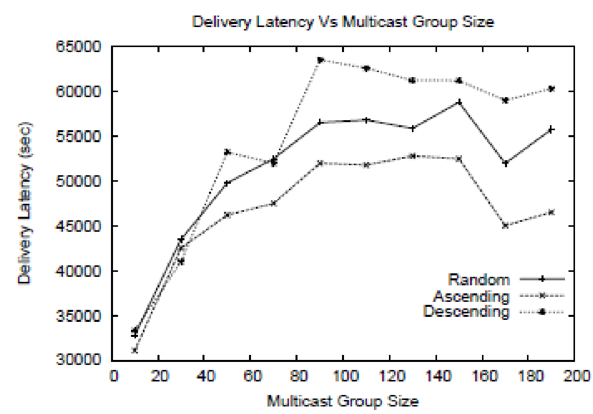


(a) $SMTBR_1(\alpha = 1.0)$

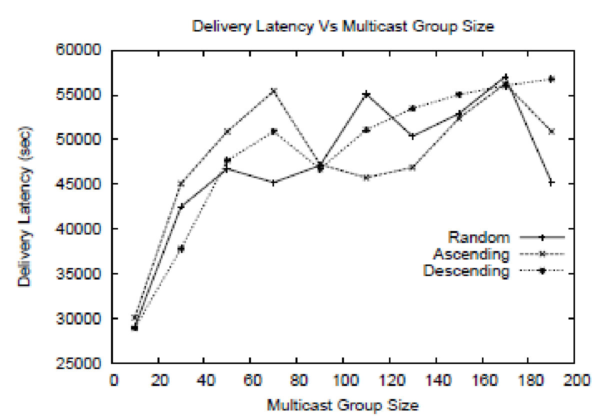


(b) $SMTBR_2(\beta = 2.0)$

Figure 8. Effect of Ordering on Message Copies



(a) $SMTBR_1(\alpha = 1.0)$



(b) $SMTBR_2(\beta = 2.0)$

Figure 9. Effect of Ordering on Delivery Latency

Figure 8(a) and 8(b) show the effect of node ordering in *SMTBR*$_1$ and *SMTBR*$_2$ specifications as the number of members in the multicast group is varied. It can be seen that in both the cases, there is an appreciable difference in the number of message copies, as the order of targets in the tree computation is changed. This can be reasoned on the fact that, based on the order in which the targets are processed, the importances of the nodes are updated. But as the importance of a node changes, the tree in the next iteration also changes. Hence the cumulative effect of the iterations is visible as the difference in the number of message copies for different node ordering.

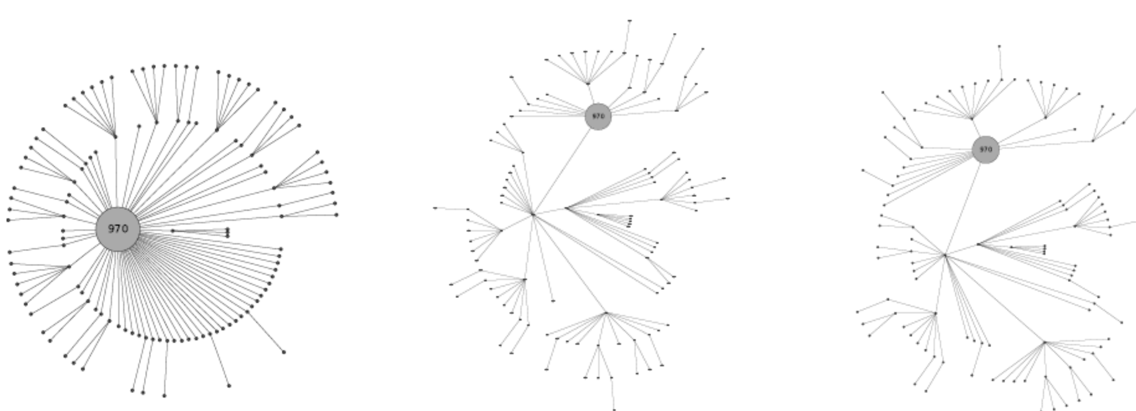
Another prominent feature of plots shown in Figure 8(a) and 8(b) is that, the number of message copies initially increases as the recipient group size is increased, but after a particular value, it starts decreasing slightly. This is because of the fact that, when the recipient group size is small, there are many intermediate nodes, which themselves are not recipients, but just act as routers. But when the group size is increased, because of importance based tree computation, more and more of the intermediate nodes that act as routers, are also the recipient nodes i.e even recipients themselves forward the messages and reduce the need for more nodes in the multicast.

5.3.2 Message Delivery Latency

The effect of target ordering while computing trees in *SMTBR* algorithm is analysed here. Figures 9(a) and 9(b) show the effect of target node ordering in *SMTBR*$_1$ and *SMTBR*$_2$ with varying recipient group size. It can be seen from these plots, that while some effect of node ordering is visible in the case of *SMTBR*$_1$, there is no such evidence for the *SMTBR*$_2$ algorithm. It can be argued that, since the importance update in *SMTBR*$_1$ is more aggressive (increasing every time, a node is selected in a path) than *SMTBR*$_2$, the deviation of the tree, with respect to latency, for different target node orderings would be lesser in *SMTBR*$_2$.

5.4. Sample Trees

Figure 10 shows a sample DTBR tree and the corresponding *SMTBR*$_1$(α = 1.0) and *SMTBR*$_2$(β = 2.0) trees, for a multicast event in SLAW mobility model with 100 recipients. The source node is shown in a big circle with the *ID* of the node. It is visually evident from the figure that number of nodes in the *DTBR* tree is reduced in *SMTBR* versions by increasing the fanout of store forward nodes. Also the depth of the tree is increased as compared to original *DTBR* tree.



(a) $DTBR$ tree (b) $SMTBR_1(\alpha = 1.0)$ tree (c) $SMTBR_2(\beta = 2.0)$ tree

Figure 10. Sample trees for a Multicast event

Table 2. Pearson's coefficient of centrality measures

σ_{xy}	Betweenness	Closeness
SMTBR₁	0.111	0.119
SMTBR₂	0.14	0.168

5.5. Centralities and "Importance"

Since the main idea of the techniques introduced in this article is to reuse the message copies that arrive at a particular node, it implicitly produces the concept of hubs in the network (as can be clearly seen in Figure 10), that cater to the delivery for many recipients. Centralities in complex graph theory is another notion to identify hubs in a large graph. Two types of centrality measures that closely resemble the requirement posed in this work are, *Betweenness* centrality and *Closeness* centrality.

Betweenness Centrality is the ratio of sum of all pair of shortest paths to the shortest paths that pass through a node, while Closeness Centrality is the reciprocal of the sum of shortest path lengths to all the other nodes from a particular node.

The Importance values for nodes are averaged out values, for 100 broadcast tree computations, with randomly selected source (all the other nodes are recipients).

Table 2 shows the Pearson's coefficient for linear correlation among the node importances and the two centrality measures. It can be seen that since the correlation coefficient is very low, there is no evidence for at least a linear dependence of the notion of Importances and the notion of centralities. This demonstrates that the centrality measures in a graph is an orthogonal concept with respect to the Importance of nodes.

VI. Conclusion

From the simulations and experiments, it can be concluded that the algorithm proposed for computation of routing trees can be used to compensate the delivery latency of the multicast events to reduce the number of participating nodes, thus reducing the number of message-copies created. From the two specifications of algorithm discussed, $SMTBR_2$ performs better than the $SMTBR_1$ version, by reducing the message-copies with lesser effect on the delivery time.

From the simulations on delivery latency and number of message copies, on the two algorithms, it can be infered that both the algorithms reach a point of saturation in number of message copies and that the relationship with delivery latency shows law of diminishing returns.

From the analysis of centrality measures with the importance of the nodes, it can be concluded that there is no obvious relationships between the concept of centralities and the notion of importances.

VII. Future Work

The present work studies the efficacy of the algorithms for two importance update functions. This study can be extended to more such versions that may be suitable for certain scenarios of multicast requirements in DTN. The algorithm proposed can also be tested with other mobility models, to study its performance in different DTN scenarios.

ACKNOWLEDGEMENTS

I would like to thank Prof. Arobinda Gupta, for his invaluable advises, suggestions and guidance, without which this work would not have been possible.

References

[1] K. Fall, "A delay-tolerant network architecture for challenged internets," *in Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '03. ACM, 2003, pp. 27–34.

[2] T. Ballardie and J. Crowcroft, "Core based tree (cbt) multicast - an analysis of multicast routing architectures," 1996.

[3] C. chuan Chiang, M. Gerla, and L. Zhang, "Forwarding group multicast protocol (fgmp) for multihop, mobile wireless networks," 1998.

[4] S. E. Deering and D. R. Cheriton, "Multicast routing in datagram internetworks and extended lans," *ACM Transactions on Computer Systems*, vol. 8, pp. 85–110, 1990.

[5] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. gung Liu, and L. Wei, "The pim architecture for wide-area multicast routing," *IEEE/ACM Transactions on Networking*, vol. 4, pp. 153–162, 1996.

[6] S. J. Lee, W. Su, and M. Gerla, "On-demand multicast routing protocol in multihop wireless mobile networks," *Mob. Netw. Appl.*, vol. 7, pp. 441–453, 2002.

[7] Z. Jin, J. Wang, S. Zhang, and Y. Shu, "Epidemic-based controlled flooding and adaptive multicast for delay tolerant networks," in *Proceedings of the 2010 Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing*, ser. UIC-ATC '10, 2010, pp. 191–194.

[8] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, ser. WDTN '05, 2005, pp. 252–259.

[9] U. Lee, S. Y. Oh, K.-W. Lee, and M. Gerla, "Relaycast: Scalable multicast routing in delay tolerant networks," in *Network Protocols,2008. ICNP 2008. IEEE International Conference on*, oct. 2008, pp. 218–227.

[10] P. Yang, "Context-aware multicast routing scheme for disruption tolerant networks."

[11] S. Symington, R. C. Durst, and K. Scott, "Custodial multicast in delay tolerant networks," in *Consumer Communications and Networking Conference*, 2007. CCNC 2007. 4th IEEE, jan. 2007, pp. 207 –211.

[12] A. Lindgren, A. Doria, and O. Schel´en, "Probabilistic routing in intermittently connected networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, no. 3, pp. 19–20, Jul. 2003.

[13] Q. Ye, L. Cheng, M. C. Chuah, and B. Davison, "Os-multicast: On-demand situation-aware multicasting in disruption tolerant networks," in Vehicular *Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, vol. 1, may 2006, pp. 96 –100.

[14] [Online]. Available: http://research.csc.ncsu.edu/netsrv/?q=content/human-mobility-models-download-tlw-slaw