Probabilistic Multicast Trees in Different Network Topology Environments

David A. Johnston, David R. McIntyre[1], Francis G. Wolff, Christos A. Papachristou

EECS Department, Case Western Reserve University
Cleveland, Ohio, USA 44106

[1] CS Department, Cleveland State University
Cleveland, Ohio, USA 44115

Abstract

Node loss and congestion changes the performance characteristics of the multicast trees. Probabilistic Multicast Trees (PMT) makes use of performance feedback, generates a probability of usage for each multicast tree based on that feedback and then makes intelligent choices about which multicast tree to use for a given packet in the presence of node loss and congestion. The advantage gained by using PMT is that it improves upon the management of the dynamic behavior of the clients where the target connectivity is constantly changing because of its feedback mechanisms and probabilistic tree selection. PMT is inserted onto three network latency topology models in a multiple tree multicast tree environment with positive improvements in the performance of data distribution.

Keywords: Network congestion, Node loss, Multicast, Application-Level Multicast, ALM, Probabilistic Multicast Trees, PMT.

1. Introduction

Node failures and network congestion in multicast networks presents unique challenges for data distribution in a wireless environment causing long delays, performance issues, and missing data. Some multicast methodologies repair the multicast trees as needed in the presence of failures. Other multicast systems improve the performance of the multicast tree through probing methods. Both methods aim to address the issue of node loss and congestion. Other multicast research has attempted to address the long delays and performance issues of node loss and congestion by using redundant paths, additional replication of data or using wholly redundant trees.

Multiple multicast trees have been shown to benefit multicasting applications and in most cases that they increase throughput and reliability. Several approaches to multiple tree multicasting have been implemented and were designed with two goals [1][2][5][13]. The first is to enhance the performance over the single multicast tree approach and the second is to handle node loss which is a fundamental problem of single multicast trees. Approaches in order to handle node loss that were built upon multiple multicasting methods include replication of packets besides just the expected distribution through the tree, forward error correction [12] and multiple description coding [7]. All of these schemes, which use additional network bandwidth, address the inherent lossy nature of wireless networks.

Probabilistic Multicast Trees [8-11] is an optimizing mechanism that is intended to improve the capabilities of any multiple multicast tree methodology with respect to management of node loss and network congestion. PMT is designed to provide two main advantages over other multiple multicast tree schemes. It improves both data delivery latency, and data delivery efficiency.

PMT achieves improved data delivery efficiency by more severely punishing trees with drop out nodes by adding increased feedback delays to these trees resulting in the overall latency feedback for any tree containing such nodes being significantly increased. An exponential moving average is used to generate the feedback value to emphasize more recent data. Trees are punished when a child node is lost. A sufficiently large penalty value is substituted for the feedback value from the lost child node. Large penalty values are applied at higher levels of a tree and have a much greater impact on the feedback latency value received by the source node. Conversely small penalty values are applied at lower levels of the tree, such as at leaf nodes, since the effect is greatly reduced because there are several tree levels between the leaf and the source.

Copyright (c) 2014 International Journal of Computer Science Issues. All Rights Reserved.

In this paper, we extend our previous work [11] on PMT in a dynamic environment by showing that PMT can be inserted on top of three widely used network latency topology models in a multiple multicast tree environment thereby suggesting its ability to improve the capabilities of *any* network latency topology model with respect to management of node loss and network congestion.

The remainder of this paper is laid out as follows: Section 2 discusses node failure and congestion, Section 3 describes the design of PMT, Section 4 describes data metrics, Section 5 shows the results of PMT versus three prominent multiple tree multicast network models, and Section 6 discusses conclusions.

2. Node Failure and Congestion Simulation

Multicasting research in the past on has focused on three main areas: building trees efficiently, reducing maintenance overhead, and using other forms besides trees to deliver the data. Multicast overlay network failures causing long delays and performance issues as the data is delivered have been only moderately addressed. Most approaches have either supported repairing the tree as failures occurred or improving the performance of the tree through probing methods. The performance improvement methods, by design, also repaired the tree. Other research focused on addressing the long delays and performance issues by using redundant paths, replicating data or using wholly redundant trees. These methodologies repair the multicast trees as needed in the presence of failures. One form of multicasting uses several multicast trees where data is sent equally on each tree. This methodology is called multiple tree multicasting. Multiple multicast trees have been shown to benefit multicasting applications in that they increase throughput and reliability and several approaches to multiple tree multicasting have been implemented. Multiple multicast trees are built at the application layer to support the data distribution. In a given multicast tree, a subset of the client nodes assists with the data delivery. With multiple multicast trees, more client nodes assist with data delivery. Packet replication, Forward Error Correction (FEC), and multiple descriptions coding (MDC) were designed to manage node loss which is a fundamental problem of single multicast trees specifically targeting wireless networks. No matter which methodology is examined, repairs of the multicast trees take a long time with respect to the time frame for data delivery where data delivery is on the order of 10s of milliseconds and tree repair is on the order of 10s of seconds.

The ideal solution would provide the following properties so that the application layer would be only minimally affected due to changes to the structure of the multicast tree [3][4][6][14][15][16]:

1. Minimize time to deliver the data.
2. Maximize the number of packets delivered.
3. Minimize bandwidth utilization.
4. Minimize network maintenance overhead.
5. Immediate detection of failures.
6. Immediate response of detected failures.
7. Non-disruptive repair mechanisms.

If there are no disruptions to the multicast overlay network, then the data is transmitted effectively and received appropriately by all client nodes via the multicast trees. However, in the presence of network congestion and node turnover, problems arise. Early multicast overlay network research investigated many of the properties of an ideal solution but fundamentally failures still cause too much delay in data delivery. Since people are the ultimate client of the data any perceived quality of experience degradation causes frustration. Although an ideal solution is not possible, any new solution should have most of the following attributes:

- Efficient Performance
- Rapid failure detection and correction
- Minimal impact to the application

The main contribution of this work is that Probabilistic Multicast Trees has been devised as an optimizing mechanism to improve the data delivery latency and data delivery efficiency of *any* network latency topology model in a multiple tree multicast tree environment in a dynamic environment. PMT was designed to be inserted onto any multiple multicasting model. The advantage gained by using PMT is that it improves upon the management of the dynamic behavior of the clients where the target connectivity is constantly changing because of its feedback mechanisms and probabilistic tree selection.

3. Probabilistic Multicast Trees

There are two reasons for using multiple trees. The first is to maintain the benefits of multiple multicast in that more nodes are actively multicasting the data. The second is to account for changing bandwidth patterns as the underlying networks exhibit their dynamic behavior. As the performance of the multicast trees change due to node loss, network congestion, tree performance improvement or other changes due to mobile nodes, the latency feedback mechanism continually provides updated latency values to the source so that as the multicast trees' probability percentage of usage is

Copyright (c) 2014 International Journal of Computer Science Issues. All Rights Reserved.

recalculated tree selection chooses the best tree most often at any given time.

Improvement due to PMT manifests itself in data delivery latency, a metric measured as an output of the process. An improvement in the metric is an indication that using PMT is advantageous. PMT is based on latency feedback which is accumulated by the parents for each multicast tree. The feedback packet consists of the averaged feedback from all the parent's children and the parent's average latency delay value. Of course, missing feedback from children causes the averaged delay value to be larger thereby penalizing the multicast tree as discussed in section 1. It is these feedback values that are used to generate the probability of usage table that the source will use to make a decision about which multicast tree to use for each packet. The latency feedback mechanism is the key to PMT.

Figure 1 illustrates three multicast spanning trees (T_1, T_2 and T_3) and their respective feedback latencies (L_1, L_2 and L_3). The source node, Src, can broadcast on three wholly separate multicast trees. In Split-Stream each tree is used in a round robin fashion to send each individual packet. For example, the first packet is sent on the blue tree, second packet is sent on the red tree, the third packet is sent on the black tree. The fourth packet will be sent on the blue tree as the process repeats until all the data is transmitted.

PMT does not follow this round robin process for tree selection. For this example, T_2 is a more efficient tree for transmission than T_1 which is more efficient than T_3. The relative probability of usage of trees T_2, T_1 and T_3 are 0.70, 0.21 and 0.09 respectively. The efficiency of each tree was measured via feedback over a period of time with the network in a steady state mode which resulted in the assigned probabilities. PMT chooses a tree for packet transmission by generating a random number between 0 and 1 and chooses T_2 if less than 0.70, T_1 if between 0.70 and 0.91 and T_3 if greater than 0.91. When the efficiency of the trees changes then the probability of usage will change based on the relative performance of each tree.

PMT is built upon the following premise: Since each multicast tree does not have the same performance characteristics PMT relies on the latency feedback mechanism from each multicast tree to generate a probability percentage of usage for each multicast tree. The probability percentage of usage for a given multicast tree is a value indicating how frequently a particular multicast tree may be chosen. For each packet sent, one multicast tree is chosen randomly based on its probability percentage of usage. The tree with the best performance will be used most often and poorer performance trees will be used less frequently. However, less frequently poorer performance trees will nonetheless occasionally be used possibly yielding improvements in latency feedback possibly due to decreased network congestion for these trees.

4. Data Metrics

Simulation data is collected and passed through a series of calculations that will be used for analysis and comparison.

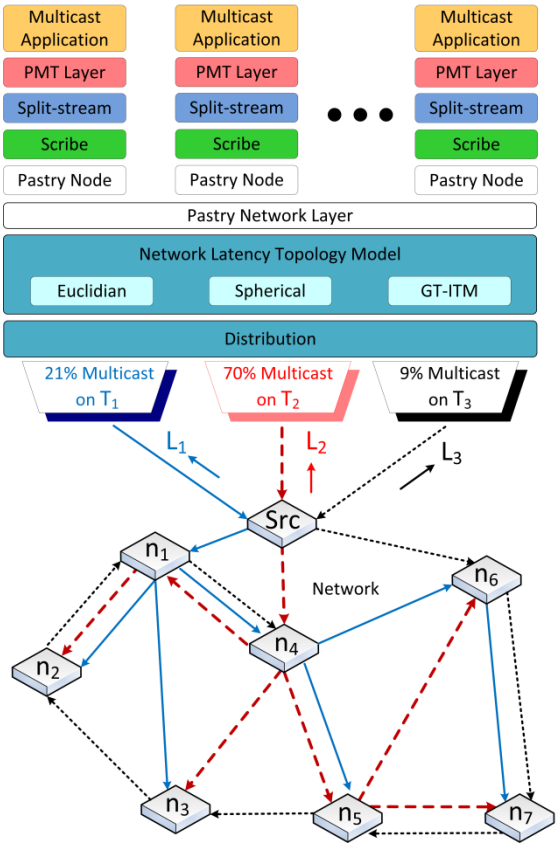

Fig. 1 PMT Multicast Tree Selection

One metric is *total delay latency time*. This is the summation of all the time differences from all the packets received. During dynamic testing some nodes are lost which means that the remaining nodes will not receive all of the packets. The *non-normalized data delivery latency* is the total delay latency time for the actual number of packets received and does not include the missing packet latency delays.

The *normalized data delivery latency* metric, *NL*, is based on the non-normalized data delivery latency

Copyright (c) 2014 International Journal of Computer Science Issues. All Rights Reserved.

metric. For each lost node a sufficiently large penalty value is substituted for the feedback value from the missing node.

5. Results

PMT was compared against three network latency topology models in a multiple tree multicast network models using Splitstream [2][3] with the FreePastry simulator [18]. FreePastry is an open source java implementation of Scribe/Pastry. This simulator provides three network latency topology models to model different aspects of the internet: Euclidean (planar) model [18], spherical model [18], and GT-ITM model [17]. Figure 2 illustrates the three network latency topology models used in our experiments.

Simulations were run as follows. The source inserts the tree number into the packet and a time stamp into each packet. The receiver uses the tree number to drive metric collections for each tree and it performs a difference calculation to generate the delay time from the source. This delay time is added to the data delivery latency total and is used for worst case delay comparison. The following enumeration describes the raw data collected by the nodes of the multiple multicast networks.

1. *The source node tracks the number of packets sent on each tree. For Splitstream this will always be the same number; however, for PMT, the number will vary for each tree.*
2. *The client node tracks the total number of packets received on each tree.*
3. *The client node tracks the worst case data delivery latency per tree.*
4. *The client node tracks the total data delivery latency for all packets received.*

PMT is compared against using the total delay latency metric. Each set of tests is averaged and the mean of the total delay latency is compared directly.

Each dynamic test simulation run had an initial node count and a specified number of nodes to be removed - approximately 10% of the total. After all the nodes were created a subset of nodes were randomly chosen for removal beginning 8 seconds after data transmission started. The chosen nodes were removed from the "active" list and placed on the "to-be- removed" list. The "to-be-removed" node list is iterated to remove the nodes at the appropriate point in the simulation so that the nodes can be removed from the trees. This process provided sufficient dynamic behavior for comparison.

5.1 Euclidean Model Dynamic Tests

The Euclidean model maps the nodes into a 2-dimensional grid. Nodes are placed randomly in the grid at start up. Packet delay calculations are based on the Euclidean distance between the two nodes.

Using the Euclidean model with 16 trees and with node counts of 550, 1100, and 2200, Figures 3 and 4 show the average total data delivery latency for the two methods. As indicated from the means of the two charts, PMT shows a small 1.5% improvement in average total data delivery latency (472000 ms versus 464000 ms).

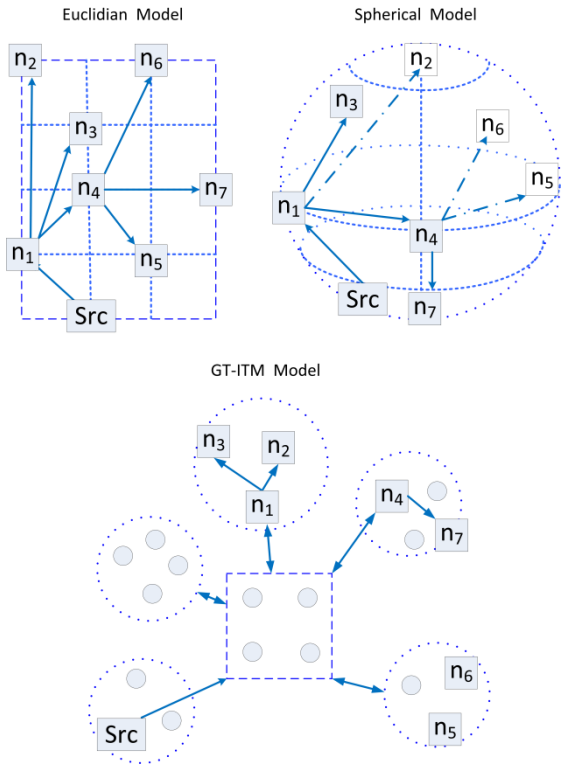

Fig. 2 Network Latency Topology Models

5.2 Spherical Model Dynamic Tests

The spherical model maps nodes randomly onto the surface of a sphere giving them spherical coordinates. Packet delay calculations are based on the chord length between the two nodes.

Copyright (c) 2014 International Journal of Computer Science Issues. All Rights Reserved.

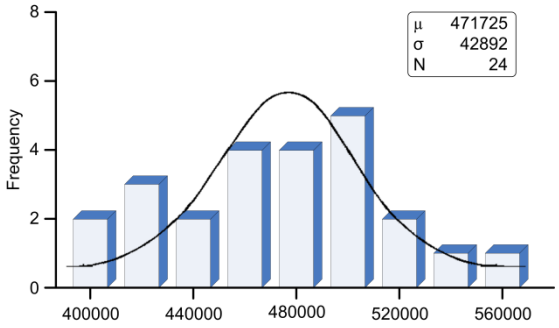

Fig. 3 Total Data Delivery Latency Averages for Splitstream, Dynamic Euclidean Model

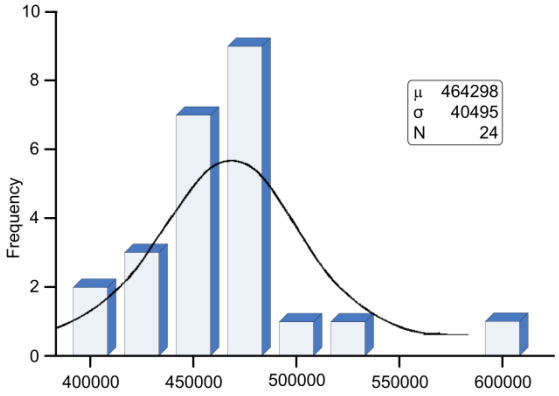

Fig. 4 Total Data Delivery Latency Averages for PMT, Dynamic Euclidean Model

Using the Spherical model with 16 trees and node counts of 550, 1100, and 2200, Figures 5 and 6 show the average total data delivery latency for the two methods. As shown by the means of the two charts, PMT shows an approximate 4.4% improvement in average total data delivery latency (341000 ms versus 326000 ms).

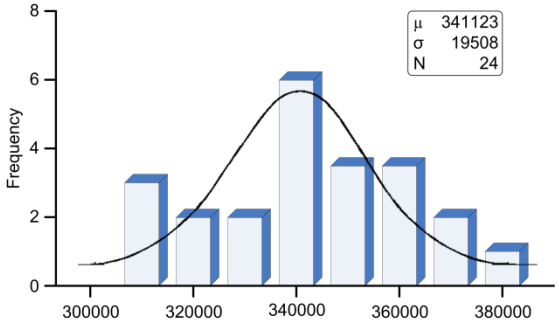

Fig. 5 Total Data Delivery Latency Averages for Splitstream, Dynamic Spherical Model

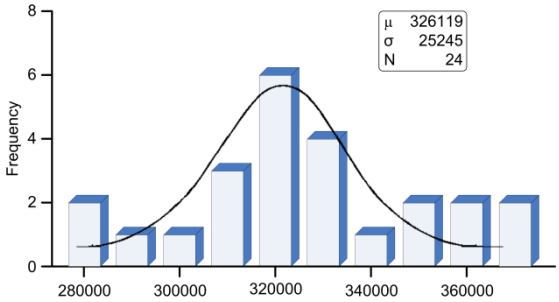

Fig. 6 Total Data Delivery Latency Averages for PMT, Dynamic Spherical Model

5.3 GT-ITM Model Dynamic Tests

GT-ITM (Georgia Tech Internet Topology Models) is an internet topology generator [17]. Since its release GT-ITM has been widely used in the scientific community for network simulations. We used the GT-ITM model with 8 trees and node counts of 550, 1100, and 2200. Figure 7 and Figure 8 show the average total data delivery latency for the two methods, PMT and Splitstream. As indicated from the means of the two charts, PMT shows a 16% improvement in average total data delivery latency. This improvement percentage is actually more impressive because the Splitstream data delivery latency was calculated using the non-normalized formula which does not include the missing packet latency delays whereas the PMT means included the estimated missing packet latency delay times.

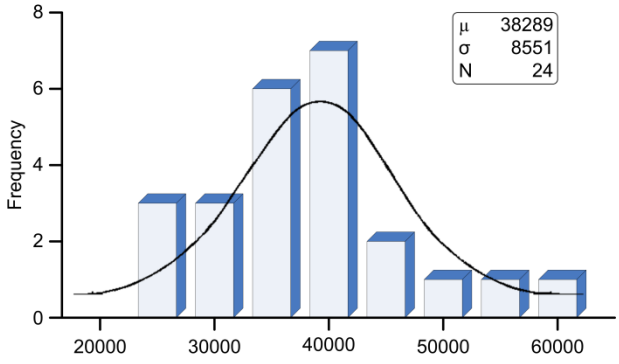

Fig. 7 PMT Data Delivery Latency Averages for Splitstream, Dynamic GT-ITM Model

Copyright (c) 2014 International Journal of Computer Science Issues. All Rights Reserved.

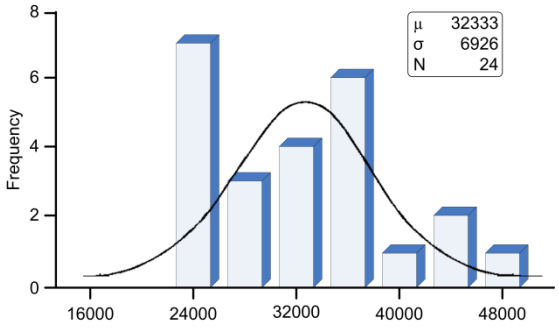

Fig. 8 PMT Data Delivery Latency Averages for PMT, Dynamic GT-ITM Model

5.4 Dynamic Tests Analysis Summary

When the tree delays are similar in size such as with the Euclidean and spherical models, the PMT algorithm provides marginal benefit with regards to the total data delivery latency. However, PMT has a consistently better response with regards to total data delivery latency between the two methods. When the tree delays are dissimilar in size, as with the GT-ITM model, then the PMT algorithms can provide better performance. However, some nodes will have little delivery improvement or even worse delivery time in spite of the PMT improvements. Since the GT-ITM model was designed to more closely model the actual Internet, we can state that the PMT system will offer delivery improvement over normal Splitstream in most circumstances even with the additional 4% overhead for feedback packets.

6. Conclusions

We have introduced an optimizing methodology, Probabilistic Multicast Trees (PMT), which was designed to be inserted onto *any* network latency topology model in a multiple tree multicast tree environment and to improve the capabilities of *any* multiple multicast tree methodology with respect management of node loss and network congestion.

Simulations with PMT on top of three network latency topology models, GT-ITM, Euclidean, and Spherical, have been shown to improve data delivery latency. As a byproduct data delivery efficiencies are improved by PMTs avoidance of trees with high node loss.

References

[1] Banerjee, S, Lee, S., Bhattacharjee, B. and Srinivasan, A., "Resilient multicast using overlays," Proc. of ACM SIGMETRICS, June 2003.

[2] Castro, M., Druschel, P., Kermarrec, A.-M., Nandi, A., Rowstron, A. and Singh, A., "Splitstream: High-bandwidth multicast in cooperative environments," Proc. of the 19th ACM symposium on Operating systems principles (SOSP '03), 2003.

[3] Castro, M., Druschel, P., Kermarrec, A.-M., Rowstron, A., "SCRIBE: a large-scale and decentralized application-level multicast infrastructure," IEEE Journal on Selected Areas in Communications, Vol. 20, No. 8, 2002, pp. 1489 - 1499.

[4] Chikarmane, Vineet and Williamson, Carey L. and Bunt, Richard B. and Mackrell, Wayne L., "Multicast support for mobile hosts using mobile IP: design issues and proposed architecture", Mobile Networks and Applications, Springer, 1998, Vol. 3, No. 4, 1998, pp. 365 - 379.

[5] Chu, Yang-hua, Rao, Sanjay G., Seshan, Srinivasan, Zhang, Hui, "A case for end system multicast," IEEE Journal on Selected Areas in Communications Vol 20, Issue 8, Oct 2002.

[6] Diot, C.; Levine, B.N.; Lyles, B.; Kassem, H.; Balensiefen, D., "Deployment issues for the IP multicast service and architecture," Network, IEEE , vol.14, no.1, pp.78 - 88, 2000.

[7] Goyal, V.K., "Multiple Description Coding: Compression meets the Network," Signal Processing Magazine, IEEE Publication Date: Sep 2001 Volume: 18, Issue 5, pp. 74-93

[8] Johnston, D. McIntyre, D., Wolff, F. and Papachristou, C., "Optimizing Application Level Multicast Trees over Wireless Networks," Proc. of the 2011 IEEE National Aerospace and Electronics Conference (NAECON'11), 2011.

[9] Johnston, D. McIntyre, D., Wolff, F., and Papachristou, C., "Probabilistic Multicast Trees," . Journal of Computer Science & Technology (JCS&T), Vol. 12, No. 1, April 2012.

[10] Johnston, D. McIntyre, D., Wolff, F., and Papachristou, C., "Data Distribution in a Wireless Environment with Migrating Nodes," . Journal of Computer Science & Technology (JCS&T), Vol. 13, No. 2, Oct. 2013.

[11] Johnston, D. McIntyre, D., Wolff, F., and Papachristou, C., "Multiple Tree Multicast in a Dynamic Environment," . International Journal of Computer Science Issues (IJCSI), Vol. 10, No. 5, Sept. 2013.

[12] Koutsonikolas, D., Hu, Y. C., "The case for FEC-based reliable multicast in wireless mesh networks," Proc. of 37th Annual International Conference on Dependable Systems and Networks, 2007 pp 491 – 501

[13] Perkins, Charles E. and Johnson, David B., "Mobility Support in IPv6", Proceedings of the Second Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), pp. 27-37 1996.

[14] Rowstron, Antony and Druschel, Peter, "Pastry: Scalable, decentralized object location and routing for large-scale peer to peer systems," Proc. of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware) 2001.

[15] Saltzer, J.H., Reed, D.P., and Clark, D.D., "End-to-end arguments in system design," M.I.T. Laboratory for Computer Science, ACM Transactions on Computer Systems, Vol 2, Issue 4, 1984, pp. 277 - 288.

[16] To, K.K., Lee, Jack Y.B., "Parallel overlays for high data-rate multicast data transfer," ACM Computer Networks 51, 2007, pp 31 - 42.

Copyright (c) 2014 International Journal of Computer Science Issues. All Rights Reserved.

[17] Zegura, E. W., Calvert, K. L. and Bhattacharjee, S. "How to Model an Internetwork", Proceedings of IEEE INFOCOM '96, San Franscisco, USA, March 1996, pp. 594 – 602.

[18] FreePastry Simulation
http://www.freepastry.org/FreePastry/

David A. Johnston received a Ph.D. in Engineering and Computer Science from Case Western Reserve University.

David R. McIntyre received a Ph.D. in Computer Science from the University of Waterloo. He is Associate Professor of Computer Science at Cleveland State University.

Francis G. Wolff received a Ph.D. from Case Western Reserve University. He is a Senior member of the IEEE and the Senior member of the ACM.

Christos A. Papachristou is Professor of Electrical Engineering and Computer Science at Case Western Reserve. He received the Ph.D. degree in Electrical Engineering and Computer Science from Johns Hopkins University. He is a Fellow of the IEEE and a member of the ACM and Sigma XI, and is listed in Who's Who in America.

Copyright (c) 2014 International Journal of Computer Science Issues. All Rights Reserved.