Data Distribution in a Wireless Environment with Migrating Nodes

David A. Johnston
EECS Department, Case Western Reserve University
Cleveland, Ohio 44106

David R. McIntyre
CS Department, Cleveland State University
Cleveland, Ohio 44101

Francis G. Wolff
EECS Department, Case Western Reserve University
Cleveland, Ohio 44106

Christos A. Papachristou
EECS Department, Case Western Reserve University
Cleveland, Ohio 44106

Abstract

The introduction of mobile wireless devices brings unique challenges for distribution of data to many devices simultaneously. An optimizing multicast methodology called Probabilistic Multicast Trees (PMT) is extended to handle mobile wireless devices. We will show that PMT multiple tree multicast system is well suited to this mobile dynamic environment.

Keywords: Mobile wireless devices, Application-Level Multicast, ALM, Probabilistic Multicast Trees, PMT, Adaptive tree selection, Content distribution.

Introduction

Smart phones, movies on demand, and other forms of streaming media information: the thirst for data access has never been greater and will only continue to grow. Multicast allows streaming media to reach multiple destinations efficiently. One technique, IP multicast, is handled directly at the hardware level by the routers themselves within the network. The globalization of streaming media presents difficulties for router based multicast especially crossing IP service provider domains due to customer charges, agreements, incompatible interfaces and permissions[6][9]. Application Level Multicast (ALM) overcome these limitations. Communication between these nodes can exploit the local underlying available local technology while globally making provider domains appear invisible [11].

The multicast overlay network can be described as a tree. Many single multicast tree solutions and multiple multicast tree solutions have been developed [1][3][5]; however, we still need to make these solutions more efficient. In particular, the dynamic behavior of multicast networks presents many challenges for data distribution in a wireless environment as shown in Figure 1. Not only do nodes join and leave a multicast network but many wireless devices are mobile and will move their location [4][9]. As wireless nodes move, the structure of the multicast tree may break, forcing it to be treated as a node failure and not movement.

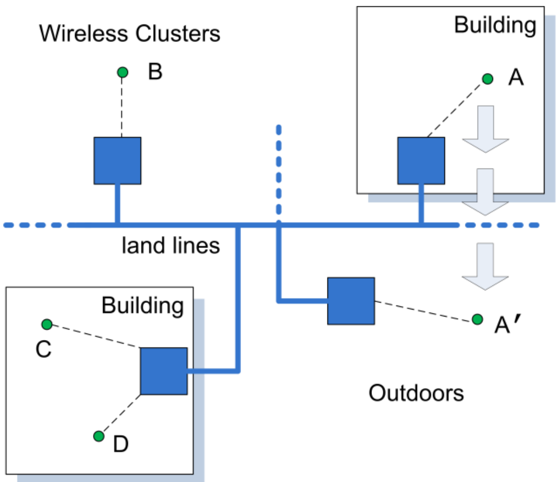

Figure 1. Mobile wireless environment

As nodes fail within a multicast tree, most multicasting approaches achieve repair by restructuring the tree. Another approach is by using probing methods. Continual probing methods will adjust the multicast tree as the nodes move and the links between them change characteristics. They improve the performance of the tree and by design repair the tree if needed [5]. In order to prevent extra overhead burden on the multicast tree, both repair and probing methods are typically performed at a much slower rate than the transmission of the data stream.

Both multicast methodologies can claim to address the issue of node mobility with their repair/improve schemes but they were not specifically designed to manage mobile devices. Multiple multicast trees show advantages over the single multicasting application for this mobile environment in that even as multicast trees change because some nodes are moving, data delivery is still relatively efficient because multiple multicast trees are being used.

We apply Probabilistic Multicast Trees (PMT) [7][8] to a mobile wireless environment. PMT is an optimizing mechanism that is intended to improve the capabilities of any multiple multicast tree methodology with respect to data delivery latency and data delivery efficiency. The design of PMT allows it to perform better than other multiple multicast tree schemes because it changes the usage of the multicast trees as the performance of the multicast trees change. The performance change for a multicast tree that contains mobile nodes could be quite substantial. Using the feedback mechanism, PMT automatically changes the tree usage pattern based on the tree performance changes caused by the mobile devices.

The remainder of this paper is laid out as follows: the mobile wireless simulation model, the PMT paradigm, the results, and finally conclusions.

The Mobile Wireless Simulation Model

We have devised Probabilistic Multicast Trees (PMT) [7][8] which is an optimizing mechanism that is designed to improve the data delivery latency and data delivery efficiency of any multiple multicast tree methodology. PMT was designed to be inserted into any multiple multicasting model. As one example of the application of PMT methodology to an existing technology, we have applied PMT to Split-stream [2]. Split-stream is a multiple multicast tree system built upon Scribe [3] and Scribe is built upon Pastry [10]. Pastry is a generic distributed hash and routing system and is a reliable routing system that delivers a message to the node whose Node ID is numerically closest to the message key. We use the FreePastry [14] simulator with the Euclidean Model for testing our wireless model. The simulator was modified to run both Split-stream alone and with PMT integrated into it. Details can be found in [8]. The Euclidean model, which is part of the FreePastry simulator, provides a two dimensional grid and the nodes are placed into this grid. The placement of nodes and calculation of path delays within the Euclidean model is modified to more closely model our wireless mobile environment. We have chosen to model a typical metropolitan community where there are many "hotspots" where wireless reception is excellent and that each hotspot is part of a wired mobile network. Clusters of nodes are grouped around these hotspots in the Euclidean grid. The Euclidean grid would normally use a strict Euclidean distance calculation to determine an "effective" delay between two nodes. This calculation is modified in an effort to model the wireless domain we have chosen and will be described below. Our wireless device model has the ability to multicast to other wireless devices. This ability to act as a multicast node can improve latency and range for a network of wireless devices. It will also reduce the dependency on a centralized wireless access point which could be a bottleneck.

We assigned 64 cluster hotspots to the 128 x 128 grid. Each cluster hotspot is given a unique 6 digit number. As nodes are created they are placed into the appropriate cluster hotspot via examination of the top six address bits of the NodeID. Each cluster has a size of 16 x 16 points in the larger grid and the wireless to wired router is assigned the origin location inside a given cluster. Some examples of cluster relay locations are {0, 0}, {32, 48} or {112, 96}.

Delay calculations will follow two separate strategies using the Euclidean distance. The first strategy is for intra-cluster delays. The second strategy is for inter-cluster delays which will include intra-cluster delays. All delays are calculated in milliseconds. Refer to Figure 1.

Four clusters shown, two are outdoors and two are building delineated by squares. There are several nodes within each cluster represented by green filled circles. Several movement/communication scenarios are illustrated. Circles A, A’ show the movement of node A from one hotspot to another denoted by A’. In one cluster node B sends a packet to another node, C, in a different cluster. The packet transmission would go from B to B’s router, then from B’s router to C’s router and then from C’s router to C. The delay time for this packet will be the sum of three items calculated as follows: Transmission #1 uses the intra-cluster calculation from B to its router. The delay time for this will be calculated as the square of the Euclidean distance between the B and the router. Transmission #2 will use cluster to cluster distance where the adjacent cluster is counted as 1 unit. For example, if the clusters are 3 clusters apart, then the delay would be 3 units. Transmission #3 from C's router to C uses the intra-cluster calculation already discussed. Note that node C can communicate to node D within the same cluster through the common router, i.e. C sends to the router which sends to D.

The reasoning behind the strategies is that any inter-cluster transmissions (transmission #2 above) are quite fast since they are all wired. The intra-cluster transmissions (transmission #1 and #3 above) are wireless and prone to packet drops and retries related to the distance between the two nodes. This causes the typical delays to be longer even for a relatively short Euclidean distance.
Node migration is managed as part of the simulation environment. As the nodes are placed into the grid, 25% of the nodes will be marked as "mobile". These nodes will have both a starting position and a destination position. The destination location is generated randomly and may be anywhere in the 128 x 128 grid. Figure 2 shows a node during part its migration. The node starts at the blue box in the upper left corner of the grid. Each second it moves toward the destination grid location for the location in the lower right corner (the black box). Each arrow indicates one time unit of movement.

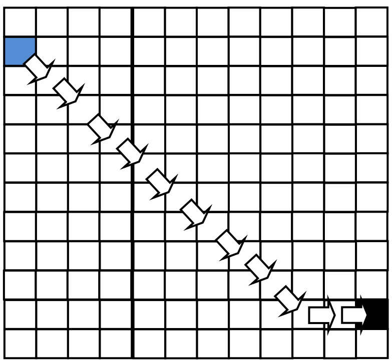

Figure 2. Migration Path for a Mobile Node

This node will arrive at its destination in 11 time units. Of course, as nodes migrate they may leave their initial hotspot and enter a neighboring hotspot.

Probabilistic Multicast Trees

PMT is based on latency feedback. Data delivery latency, L_d, is the summation of all the source-to-destination packet delivery times. In order to provide latency feedback a separate periodic thread was created that executes at a fixed time period of one second. This thread sends feedback data to its parent 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. New feedback values overwrite older feedback values. 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 Scribe [3] "anycast" functionality was added to enable this feedback from child to parent. The latency feedback mechanism is the key to PMT.

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 higher a value for a particular multicast tree, the higher its probability is for being chosen for the next packet to be sent. As a result, 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.

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. The decision to select a multicast tree for a packet about to be sent is based on the generation of a random number and this number is applied against the trees' probability percentage of usage to make the selection. 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 recalculated tree selection chooses the best tree most often at any given time. Recalculation is performed at regular intervals and this interval is once per second.

PMT improves upon the management of the dynamic behavior of the clients when the target connectivity is constantly changing because of its feedback mechanisms and probabilistic tree selection. This improvement 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.

Figure 3 illustrates three multicast spanning trees. To the source node each tree is a wholly separate multicast tree. 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, Tree *2* has been determined to be a more efficient tree for transmission than Tree *1*. Tree *1* has been determined to be a more efficient tree for transmission than Tree *3*. Tree *2* is assigned a probability of usage of 0.67 based on its relative efficiency as compared to the other two trees. Tree *1* is assigned a probability of usage of 0.31 based on the same criteria. Tree *3* is assigned a probability of usage of 0.02. 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. The calculation of the probabilities will be described below. To choose a tree for transmission a random number is generated. If the random number is less than 0.67 then Tree *2* is chosen. If the random number is between 0.67 and 0.98 then Tree *1* is chosen. If the random number is greater than 0.98 then Tree *3* is chosen. This process is repeated for each packet transmitted. As long as no significant changes occur in the performance of the trees, then the probability of usage for each tree will remain the same. When the efficiency of the trees changes then the probability of usage will change based on the relative performance of each tree.

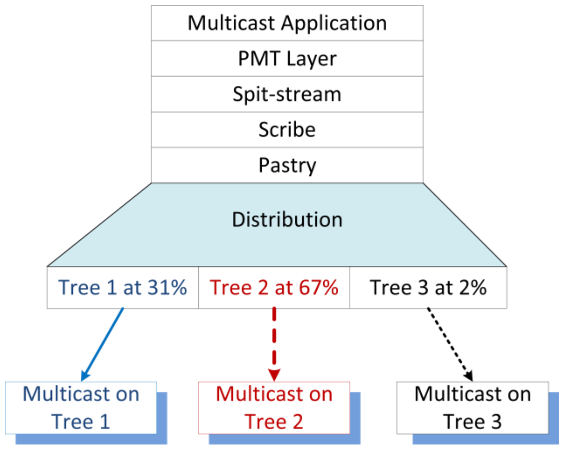

Figure 3. PMT Multicast Tree Selection

Results

Each test consists of one simulator run which sends 2048 packets through the multicast network. For PMT and Split-stream, the effective packet rate is the same. For PMT tests packets were sent at 40 millisecond intervals. For Split-stream tests, packets were sent in groups based on the number of multicast trees. For 4 multicast trees, a group of 4 packets was sent at 160 millisecond intervals. Tests were run 24 times with each combination of 500 and 1000 nodes. The tests were run first with unmodified Split-stream code. Then each test was repeated with PMT code.

For the migration runs the mobile nodes started moving at 20 seconds into the simulation and then moved once a second for up to 40 seconds before stopping. The nodes traversed a steady path from starting position to the destination position during these 40 seconds. A node may or may not arrive at its destination position before the 40 seconds have elapsed.

Figures 4 through 7 show the data delivery latency results for the migration suite of tests for PMT and Split-stream. The mean data delivery latency migration results for PMT and Split-stream are summarized in Table 1. Also the mean data delivery latency results for static (non-migration) are summarized in Table 2.

Node Count	PMT results	Split-stream results	% Improvement
500	921000	1124000	18%
1000	1047000	1217000	14%

Table 1 Mean Data Delivery Latency Migration test results summary

Node Count	PMT results	Split-stream results	% Improvement
500	383000	441000	13%
1000	403000	429000	6%

Table 2 Mean Data Delivery Latency Static test results summary

Table 1 shows that for migration PMT is more efficient at data delivery latency when compared to Split-stream by an average of 14% to 18%. Typically PMT will be more efficient by about 16%. This is based on the random sample set of tests that were run with PMT and Split-stream. This is a significant improvement. Table 2 shows that for static nodes PMT is again more efficient at data delivery latency when compared to Split-stream by an average of 6% to 13%, the smaller improvements being due to the ability of PMT to learn the changing nature of migrating nodes not being utilized in a static environment. If we compare the overall migration results for both PMT and Split-stream a major difference stands out. The total latency is significantly higher in the case of migrating nodes as compared to static nodes. The reason is that migrating nodes greatly increases the delays for all nodes. The delays will vary greatly from test to test because the destination grid locations of the migrating nodes are determined randomly.

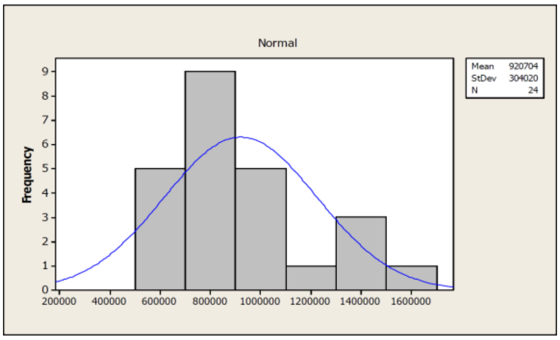

Figure 4 Migration PMT data delivery latency for 500 nodes

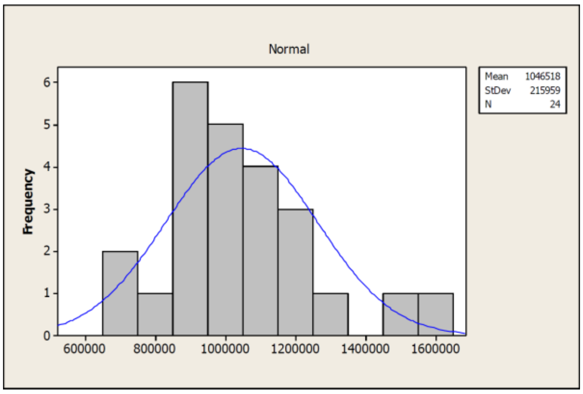

Figure 5 Migration PMT data delivery latency for 1000 nodes

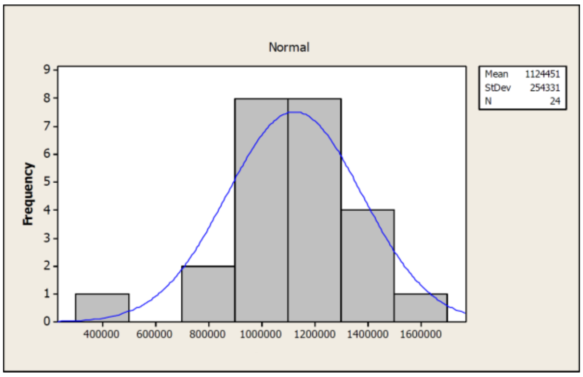

Figure 6 Migration Split-stream data delivery latency for 500 nodes

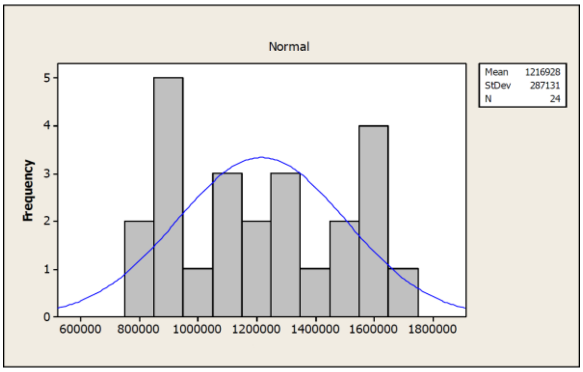

Figure 7 Migration Split-stream data delivery latency for 1000 nodes

Conclusion

The need for information is increasing at an ever expanding rate. The rate of increase will continue to put a strain on our current network hardware resources. There is a need to embed additional intelligence into our Software applications to help make up for the bandwidth shortfall and to manage the dynamic behavior of our wireless systems.

Multiple tree multicasting has been shown to improve performance over single multicast tree protocols. We have shown that PMT built on top of an existing multiple multicast tree protocol shows a further improvement for data delivery latency over the base protocol in a simulated wireless environment with migrating nodes. An improvement of 16% is significant in the decrease in latency allowing the use of the network bandwidth for other applications.

We have introduced the Probabilistic Multicast Trees paradigm which can be built into any existing multicast tree protocol. The addition of feedback and random tree selection with PMT reduces data delivery latency in a wireless node migration environment. Feedback results in better trees being used more frequently which results in a reduction of latency. At the same time, the occasional selection of poor trees allows PMT to detect improvements in such trees possibly due to congestion improvements. As tree performance changes and feedback data reflects in the probability of usage table, PMT learns which trees are better at any given time and can make fuller use of them. Additionally as node failures cause more of a negative probability of usage impact to a given tree, PMT will use such trees less frequently. This self adjusting behavior drives the improvement delivered by PMT in an environment of migrating wireless nodes.

Bibliography

[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 20 (8) (2002) 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] 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).

[8] 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.

[9] 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 1996), pp. 27-37 1996.

[10] 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.

[11] 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.

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

[13] Zakhor, Avideh, Wei, Wei, Multiple Tree Video Multicast over Wireless Ad Hoc Networks, Proc. of IEEE International Conference on Image Processing, Oct. 2006, pp 1665 - 1668

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

Received: June 2013. Accepted: August 2013.