How Bad Are Overlay Networks?

Stephen Cogdon and Ian Wakeman
University of Sussex

Abstract

This paper assesses the incorporating of candidate nodes into overlay multicast trees. Overlay networks attempt to compensate for the awkwardness and inefficiencies that can exist across the Internet for group communication. This work uses a network simulation application to assess the feasibility of overlay networks and presents the results found. Results show that there are persistent improvements using the metric of cost between the optimal KMB tree and the minimum path spanning tree and that there are significant improvements, again using the metric of cost, between n-way distribution without a centroid node and n-way distribution with a centroid node of approximately twenty per cent in each instance. This is to provide an application layer active networking infrastructure for distributed virtual reality.

1 Introduction

The Internet is a collection of independent networks. These individual networks communicate with one another using various gateway protocols [1]. IP multicast [2] can be run over these individual networks, but communication across borders or gateways has produced technicalities. Overlay networks attempt to compensate for the awkwardness and inefficiencies that can exist across the Internet.

The Mbone [3] is a virtual network for audio and video transmissions using the IP multicast protocol distance vector multicast routing protocol (DVMRP) [4]. The 6-bone is an experimental network for testing IPv6 [5]. The X-bone is an experimental overlay network configuration tool used for the deployment of such virtual networks as the Mbone and 6-bone [6].

Tree building protocols configure and self-organise the overlay network. Tree building control protocol (TBCP) is a constrained tree building technique [7] signifying that the number of children a parent node can have is controlled by the root node. When a node joins a group it is upto them to discover an appropriate nearby parent. TBCP is generic and is designed to build overlay spanning trees among participants of a multicast session, without any specific help from the network routers.

Overcast is an unconstrained tree building technique [8] signifying that the number of children a parent node can have is not controlled by the root node. Again, when a node joins a group it is upto them to discover an appropriate nearby parent. Overcast provides scalable and reliable single-source multicast for large-scale applications using a simple protocol for building efficient data distribution trees that adapt to changing network conditions.

End system multicast is an alternative architecture for small and sparse groups, where end systems implement all multicast related functionality including membership management and packet replication [9]. The Narada protocol allows end systems to self-organise into an overlay structure using a fully distributed protocol. It is multi-source multicast for small-scale applications and uses the approach of initially building the mesh from which the tree is then developed.

Yoid is a generic architecture that attempts to take reliable and asynchronous distribution from the server-based track, and dynamic auto-configuration via a simple API from the IP multicast track [10]. It allows a group of endhosts to auto-configure themselves into a tunneled topology for the purpose of content distribution. When a node joins a group it is upto them to discover an appropriate nearby parent.

A resilient overlay network (RON) is an architecture that allows distributed Internet applications to detect and recover from path outages and periods of degraded performance within several seconds [11]. The RON nodes monitor the functioning and quality of the Internet paths among themselves, and use this information to decide whether to route packets directly over the Internet or by way of other RON nodes, optimising application-specific routing metrics.

2 Experimental Design

2.1 Optimal Distribution Trees

The minimum path spanning tree is a tree of the graph that encompasses all the nodes of that graph by using as few edges as possible [12]. One algorithm that attempts to produce optimal distribution trees is known as the KMB approximation algorithm after the researchers who designed it, Kou, Markowsky and Berman. The heuristic algorithm has a worst case time complexity of $O(|S||V|^2)$ and it guarantees to output a tree that spans a set S, where S is a subset of the vertices of a set V, with total distance on its edges no more than $2(1 - \frac{1}{l})$ times that of the optimal tree, where l is the number of leaves in the optimal tree. The KMB algorithm consists of the following five steps [13].

The unicast delivery of a data packet for each group member. A given sender will have a record of the shortest path from itself to each individual group member.

2.2 Hierarchical Configuration

Hierarchical graphs would assume connection to external networks. An example may be the Internet, a collection of interconnected networks, although due to the continually changing and dynamic structure of the Internet hierarchical graphs may not resemble this structure but are considered appropriate for the generally recognised ideal of an internet [14].

The transit-stub model produces hierarchical graphs by composing interconnected transit and stub domains [14]. A connected random graph is constructed, each node in the graph represents an entire transit domain. Each node in the graph is then replaced by another connected random graph, representing the backbone topology of one transit domain. For each node in each transit domain a number of connected random graphs are generated representing the stub domains attached to that node. Finally, a number of additional edges between pairs of nodes, one from a transit domain and one from a stub, or one from each of two different stub domains are added.

2.3 Metrics

The metrics of an experiment are the measures used for analysing the output. Cost, delay and scalability are measures which are recognised in network simulation [15]. Analysing the cost and the delay of a network and its internal operations appear to be standard measurements used in much of the literature conducting network simulations [16, 17].

The cost of a multicast tree is the sum of the weights of all the links in the tree [15]. A good multicast tree tries to minimise this cost. The end-to-end delay from the source node to the destination node is the sum of the individual link delays along the route [15]. A good multicast tree tries to minimise the end-to-end delay for every source-destination pair in the group. The scalability of an algorithm as the group size or topology size increases is important. Constructing a multicast tree for a large group should require reasonable amounts of time and resources when considering scalability [15].

3 Simulation

The experiment has been carried out using a network simulation application implemented in Java [18] and using a set of graph foundation classes available for download [19]. The application reads in a network topology as given by the Stanford GraphBase and its alt format [14] parsing it into the graph data structure used by the Java graph foundation classes. This graph can then be displayed visually and a button is available for beginning a tree comparison.

Upon the user clicking this button a set of random group members are selected and the optimal KMB tree between the group members is calculated and displayed. The three sets of statistics available represent the cost and delay figures for the KMB tree, the minimum spanning tree between the group members and n-way distribution.

As defined in the previous section the cost represents the number of links necessary in each one of the three cases and the delay shows the number of nodes the data packets will travel through that separate the

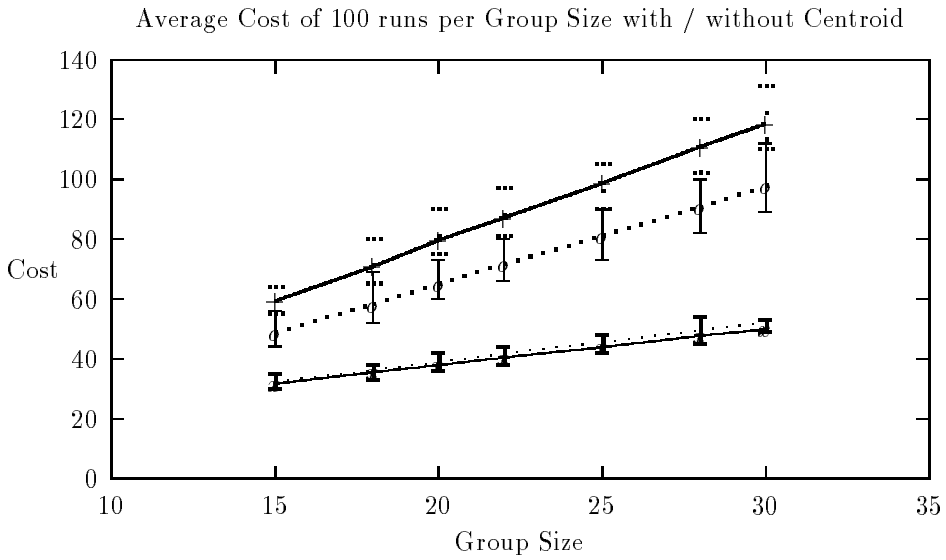
group members. To efficiently run this experiment the application has been automated removing its usability, but providing the author with a large set of data for assessing the acquired results.

3.1 Results

A 100 node topology was used with a random set of an average 180 edge connections. A 150 node topology was used with a random set of an average 250 edge connections. A 200 node topology was used with a random set of an average 320 edge connections. A 300 node topology was used with a random set of an average 520 edge connections. These were all used with a hierarchical transit-stub graph configuration. The random set of edge connections are determined using standard probability functions [14] and ensure all nodes are connected at least once to another node.

For each n node topology 100 runs were conducted for each group size without the centroid present and again with the centroid. The group members remained the same for a run firstly without the centroid and then with the centroid. The figures available for the KMB tree and the spanning tree in both instances remained constant. For the 100 runs ten different transit-stub topologies were used and ten runs were conducted on each one. This was to ensure randomness and variety in the transit-stub topologies.

The graph below measures the cost in the number of links used to construct either the KMB tree, the spanning tree or the accumulation of shortest paths that represent the n-way distribution to form the paths between the randomly selected group members for each individual group size. It can be seen that there are persistent improvements between the optimal KMB tree and the minimum path spanning tree. It can also be seen that there are significant improvements between n-way distribution without a centroid node and n-way distribution with a centroid node of approximately twenty per cent in each instance.



KMB Tree - x .. Spanning Tree - . .. N-Way without Centroid - $+$ .. N-Way with Centroid - o
Figure 1. 100 node transit-stub graph.

4 Conclusion

This paper has assessed the incorporating of candidate nodes into overlay multicast trees. Further work will be to design a protocol for tree building and group communication which will be assessed initially in network simulation. This will provide the basis for further intended experiments in clock synchronisation and load balancing relevant to providing an application layer active networking infrastructure for distributed virtual reality.

References

[1] Y. Rekhter and T. Li. A border gateway protocol 4 (bgp-4). *Internet Engineering Task Force, RFC 1771.*, 1995.

[2] S. Deering. Host extensions for ip multicasting. *RFC 1112, August.*, 1989.

[3] M. R. Macedonia and D. P. Brutzman. Mbone provides audio and video across the internet. *IEEE Computer, 27(4), pp. 30-36.*, 1994.

[4] C. Partridge, D. Waitzman, and S. Deering. Distance vector multicast routing protocol. *RFC 1075, November.*, 1988.

[5] 6Bone. 6bone web pages. *http://www.6bone.net/, December.*, 2001.

[6] J. Touch. Dynamic internet overlay deployment and management using the x-bone. *In Proceedings of the ICNP. pp. 59-68.*, 2000.

[7] L. Mathy, R. Canonico, and D. Hutchison. An overlay tree building control protocol. *In Proceedings of International Workshop on Networked Group Communication (NGC), London.*, 2001.

[8] J. Jannotti, D. Gifford, K. Johnson, F. Kaashoek, and J. O'Toole. Overcast: Reliable multicasting with an overlay network. *In USENIX OSDI 2000, San Diego, California, October.*, 2000.

[9] Y-H. Chu, S. Rao, S. Seshan, and H. Zhang. Enabling conferencing applications on the internet using an overlay multicast architecture. *In Proceedings of ACM Sigcomm, San Diego, California.*, 2001.

[10] P. Francis. Yoid: Extending the internet multicast architecture. *Technical Report, ACIRI, April. http://www.aciri.org/yoid.*, 2000.

[11] D. Anderson, H. Balakrishnan, M. Frans Kaashoek, and R. Morris. Resilient overlay networks. *In Proceedings of the 18th ACM SOSP, Banff, Canada.*, 2001.

[12] A. Aho, J. Hopcroft, and J. Ullman. *Data Structures and Algorithms.* Addison-Wesley., 1983.

[13] L. Kou, G. Markowsky, and L. Berman. A fast algorithm for steiner trees. *Acta Informatica, Vol. 15, pp. 141-145.*, 1981.

[14] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee. How to model an internetwork. *Proceedings of the IEEE Infocomm.*, 1996.

[15] L. Sahasrabuddhe and B. Mukherjee. Multicast routing algorithms and protocols: A tutorial. *IEEE Network, pp. 90-102, January.*, 2000.

[16] M. Doar and I. Leslie. How bad is naive multicast routing. *Proceedings of the IEEE Infocomm.*, 1993.

[17] L. Wei and D. Estrin. The trade-offs of multicast trees and algorithms. *In International Conference on Computer Communications and Networks.*, 1994.

[18] J. Gosling, B. Joy, and G. Steele. *The Java Language Specification.* Addison-Wesley. http://java.sun.com/docs/books/jls/index.html., 1996.

[19] Java. Java graph foundation classes. *http://www.alphaworks.ibm.com/tech/gfc, August.*, 2001.