Application-Level Multicast for a Skype Clone

Stephen Cogdon
Pathfinder Projects Limited

Abstract

This paper speculates on the integration of application-level multicast with a Skype clone, an application that implements Voice over Internet Protocol (VoIP). Skype is a globally successful application of IP telephony using a peer-to-peer network. Researchers have shown how efficiently application-level multicast can be adopted by a peer-to-peer network, I now wish to speculate on how well a Skype clone would perform through the use of application-level multicast.

1. Introduction

Following over a decade of research IP multicast has still not been comprehensively deployed throughout the Internet [1]. Recent years have seen the innovation of application-level multicast as researchers attempt to reduce the complexities of multicast by moving it away from the network-level and implementing it at the application-level [2]. Application-level multicast is seen as a viable alternative providing adaptability, hierarchy and scalability to end hosts. Application-specific overlay networks can be built according to the application semantics providing accessible functionality to the end host and achieving greater efficiency and performance than IP multicast [3].

2. Cluster Tree

2.1 Overview

I use the tree building protocol presented in [3], a scalable overlay cluster tree, to represent the peer-to-peer network to be used for this work and designed with application-level multicast in mind. The work first explained the need for a programmable node in the network that can act as a root for maintaining group configuration and efficient tree building. It then showed the benefit of incorporating a programmable node into the network achieving a greater efficiency of approaching 20% when compared to a group configuration without a root node. The work introduced the cluster tree as an implementation of a tree building protocol using a programmable node. It then compared the cluster tree with a nearest-neighbour join naive tree and show a cost improvement of approaching 10% and a scalability improvement of approaching 40%.

The work analysed the efficiency metrics for an overlay network and showed that the cluster tree has a maximum delay improvement of approaching 65% over the naive tree, but in contrast, the naive tree has a maximum node degree improvement of approaching 75% over the cluster tree. This leads us to acknowledge that the cluster tree is built breadth-first, whilst the naïve tree is built depth-first and, in both cases, the tree balances out as the group size increases.

In providing functionality to the cluster tree the work showed that in comparison to the efficiency of a minimum spanning tree the cluster tree can provide scoped multicast to approaching 40% of the group. These results allowed us to present an application-level solution providing application-specific features dependent upon the application semantics.

2.2 Logical Design

As mentioned, a suitable tree building implementation of the programmable node technique is one that provides clustering of group nodes when these nodes are located in close proximity to one another [4]. The tree is scalable, adaptive and hierarchical providing clustering techniques either to be used as a protocol within its own right or for extending an existing tree building protocol such as tree building control protocol (TBCP) [5] when dealing with clusters of nodes that exist together in a given region of the network.

The cluster tree uses a root node for initial communication with a joining node. The new node is assumed to know the location of the root node and measures its distance metric of round-trip time between itself and the node by pinging the root. The new node then sends a join request message to the root node together with the value of the round-trip time. Given this round-trip time value the root node will return the list of its children that are located in relative proximity to the new node.
The root achieves this by establishing boundaries or ranges in which its children are positioned based on each of their round-trip time values. The boundaries are termed as levels and are defined as follows. Based on the values of round-trip time measurements, or dist, level 0 is defined in the range **$0 < dist \le 1$ (1)** and level i is defined in the range **$(1 + \Delta)^{i-1} < dist \le (1 + \Delta)^{i}$ (2)**, with Δ greater than 0 and the value of Δ generally varying in unit steps from 1 to 5. This leads to an exponential distribution in the boundaries as the levels increase.
All nodes in the tree, including the root, are known as cluster heads as they are all potential parents to many child nodes. Each node in the tree, apart from the root, is therefore associated with a given radius value with which to attract child nodes. This radius value for a given level i is defined as radius i in the range **$((1 + \Delta)^{i} - (1 + \Delta)^{i-1}) / 2$ (3)**.
Thus, when the root node receives a join request from a new node it returns the list of children it has corresponding to the level in which the round-trip time value falls. The new node then measures the round-trip time between itself and each one of the child nodes it now has knowledge of. If any of these distance values are less than the given radius value then the new node will repeat the joining process with the child node that it is closest to in terms of round-trip time values.
This process will continue until eventually the new node reaches a cluster head it is closest to and that has no child nodes. The new node becomes a first child of that cluster head and itself becomes a singleton cluster awaiting its own child nodes. The procedure is recursive as each cluster head has its own set of defined levels and radius values. If a round-trip time value from a new node falls within a radius value of a cluster head then this cluster is said to be an attracting cluster.

2.3 Theoretical Design

A theoretical analysis of the cluster tree has been undertaken and has provided a number of optimisations [6]. The notion of levels has been removed to ensure these boundaries do not restrict nodes that may have existed in close proximity on either side of the boundary, but unable to have knowledge of one another or make any communication.
The analysis has used a Cartesian coordinate application space to ensure precise values to represent the round-trip time estimate between two nodes. The radius value remains and is now defined as r / 2 which is further defined as a node's distance value to its parent divided by two and termed its sphere of influence. So, a cluster head's sphere of influence is r / 2 where r is the distance between the node and its parent. When a joining node sends the root or a cluster head the distance r between them it receives the list of child nodes that are within the boundaries of 2r and 2r / 3.
The new node will follow the same procedure and measure its distance value to each of the returned child nodes that fall within these less restrictive boundaries. If a distance value falls within a node's sphere of influence then the new node will send a join request to the node with the sphere of influence the new node falls within and with the minimum distance. This process is repeated recursively until a new node has found an existing cluster head with no children to access.

3. Skype

3.1 Overview

VoIP is providing the Internet with one of its most significant commercial uses which will continue to revolutionise the way we communicate with the world around us. One such implementation of VoIP is Skype, a program that can be downloaded and installed for free [7]. The program is a Skype client that allows you to contact someone via their Skype client for free wherever they maybe located within the world.
Whereas communication between Skype clients is free, interaction between a Skype client and a normal telephone is chargeable. SkypeOut allows a Skype client to contact a non-Skype user on a

normal telephone, SkypeIn allows a non-Skype user to contact a Skype client from a normal telephone and Skype Voicemail allows a non-Skype user to leave a message at a Skype client from a normal telephone. Skype Chat provides one-to-many communication with, at present, up to forty-eight other people able to join in and perform a group chat. A Skype client can make a conference call with, at present, up to four other people.

Skype Video Calling provides one-to-one video conversation and also offers the platform for videoconferencing. To aid a group Skype provides a secure and encrypted file transfer option that can handle files of any size. Skype does not consider itself as a replacement for the telephone network and this maybe due to the rules and regulations governing emergency numbers. Skype can be seen more as a piece of groupware which aids users in their collaboration and communication whilst they are located remotely.

3.2 Properties

Skype is a peer-to-peer network consisting only of Skype clients [8]. A Skype client has a host cache that contains the IP addresses of a number of super nodes. A super node allows the Skype client to connect to the Skype network each time the user logs in. Communication between Skype clients is then carried out through the Skype network. This allows for efficient and optimal communication as Skype clients can keep one another informed of congested routes and can provide alternate, less dense paths to traverse instead.

Recent work has recorded 3.9 million users online on the Skype network at one time and in one experiment 250K supernodes were discovered [9]. The work establishes the geographic distribution of the supernodes as 45-60% in Europe, 15-25% in North America and 20-25% in Asia. The Skype network is at its busiest during working hours with nodes joining in the morning and leaving in the evening. Of course, geographical location balances out usage as working hours alternate between Europe, North America and Asia.

4. Design

4.1 Application-Level Multicast Implementation

The cluster tree provides an application-specific overlay network infrastructure according to the application semantics for an application. It can be considered that an application-specific overlay network can be adapted to suit the requirements of applications as diverse as a distributed virtual environment and a sensor network. The cluster tree can show how application-level implementations of multicast can actually be more efficient for the network in certain application scenarios.

The performance comparison would be the effectiveness of the application-level implementation against its IP counterpart and it could be established as to whether the benefit of the more accessible and manipulable application-level implementation outweighed any significant reduction in performance it may have against its IP counterpart. The intention is to simulate an IP multicast protocol such as DVMRP [10] or PIM-SM [11] at the application-level. This would form the back-end communication whilst the front-end interface would take on a Skype appearance allowing users to start, join and participate in group chats or conference calls.

4.2 Skype Clone Integration

The core components of a cluster tree installation could be downloaded initially with applications, such as a Skype clone, being accessed upon requirement by a node. Integration requires the implementation of a Skype clone with instances for the use of application-level multicast with a Skype clone being group chat, conference calling and video calling. As explained, the back-end of the implementation would be concerned with application-level multicast whilst the front-end would provide the user interface, collect the relevant data for sending and feedback to the user data received from the network.

In initiating a Skype chat or conference call this would set-up the appropriate routing information for distribution across the nodes in the tree. All paths should be reached in log (n) steps as described in detail through the work carried out in designing Chord [12]. Data can then be sent and received by users across the tree via the nodes routing the multicast paths using the appropriate protocols such as RTP [13] and TCP [14].

5. Conclusion

There are a number of multi-user applications that are continuing to evolve for which IP multicast cannot provide all the features necessary to satisfy the requirements for, and with each individual application likely to vary in its specific needs, IP multicast is unable, and was never intended, to provide a generic solution. Thus, application-level multicast should be seen as a feasible solution in dealing with the precise behaviour and semantics of an application at the expense of a small reduction in network efficiency in comparison with IP multicast.

Multi-user applications such as distributed virtual environments, Instant Messaging, conference calling and video conferencing all differ in their specific behaviours and requirements. With application-level multicast the tree controlling the distribution of information between the user group can be designed and built with the application semantics in mind, and appropriate functionality can be tailored and applied accordingly.

It is essential then to ensure the appropriate underlying infrastructure is in place to deal with, and adapt to, the requirements of these various applications. Application-level multicast is seen as a solution to meet the requirements of these applications, and if the initiative and momentum of the present research output is sustained, then application-level multicast is likely to become as important an ingredient to the Internet as IP multicast is which would then promise a very challenging, but exciting time for the designers and the users of group communication applications alike.

References

[1] S. Deering. Host extensions for {IP} multicasting. RFC 1112, August 1989.

[2] Y-H. Chu, S. Rao, and H. Zhang. A case for end system multicast. ACM SIGMETRICS Performance Evaluation Review, Vol. 28, No. 1, pp. 1-12, June 2000.

[3] S. Cogdon. Application-level multicast for group communication. PhD Thesis, Department of Informatics, University of Sussex, Brighton, December 2003.

[4] L. Mathy, R. Canonico, S. Simpson, and D. Hutchison. Scalable adaptive hierarchical clustering. IEEE Communications Letters, Vol. 6, No. 3, pp. 117-119, March 2002.

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

[6] I. Wakeman and L. Mathy. A theoretical analysis of the cluster tree algorithm. Internal Letter, Department of Informatics, University of Sussex, Brighton, February 2002.

[7] Skype. Skype web pages. http://www.skype.com/, June 2006.

[8] S. Baset and H. Schulzrinne. An analysis of the Skype peer-to-peer Internet telephony protocol. In Proceedings of the 25th IEEE Infocom, Barcelona, Spain, April 2006.

[9] S. Guha, N. Daswani, and R. Jain. An experimental study of the Skype peer-to-peer VoIP system. In Proceedings of the Fifth International Workshop on Peer-to-Peer Systems (IPTPS), February 2006.

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

[11] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, and L. Wei. The PIM architecture for wide-area multicast routing. IEEE/ACM Transactions on Networking, Vol. 4, No. 2, pp. 153-162, April 1996.

[12] I. Stoica, R. Morris, D. Karger, M. Frans Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. ACM SIGCOMM Computer Communication Review, Vol. 31, No. 4, pp. 149-160, October 2001.

[13] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A transport protocol for real-time applications. RFC 1889, January 1996.

[14] J. Postel. Transmission control protocol. RFC 793, September 1981.