Enhancing the multicast performance of structured P2P overlay in supporting Massively Multiplayer Online Games

X. Jiang[*] F. Safaei[†]
P. Boustead[‡]

[*]University of Wollongong
[†]University of Wollongong, farzad_safaei@uow.edu.au
[‡]University of Wollongong

This conference paper was originally published as Jiang, X, Safaei, F, Boustead, P, Enhancing the multicast performance of structured P2P overlay in supporting Massively Multiplayer Online Games, 15th IEEE International Conference on Networks ICON 2007, 19-21 Nov 124-129.

Enhancing the multicast performance of structured P2P overlay in supporting Massively Multiplayer Online Games

Xinbo Jiang, Farzad Safaei, Paul Boustead
Smart Internet Technology CRC,
Telecommunications and Information Technology Research Institute,
University of Wollongong, Australia
xinbo@titr.uow.edu.au, farzad@uow.edu.au, paul@titr.uow.edu.au

Abstract*-Scribe is a scalable application level multicast infrastructure. We have developed two techniques to improve the performance of Scribe in terms of latency and bandwidth distribution. The first technique identifies that the final hop of Scribe traffic path is largely selected without any proximity consideration and incurs the longest distance traveled. To overcome this, we introduce Proximity Neighbor Selection (PNS) into the final hop for latency improvement. The second technique builds a hierarchical two-level overlay. While PNS can be applied at both levels for latency performance, the bandwidth stress required by applications can now be distributed among the nodes in the higher level overlay. Our simulation using GT-ITM topology has shown that both techniques have improved the latency performance for more than 30 percent, and the two-level overlay has improved the bandwidth distribution up to 2.7 times, comparing with what can be achieved by a standard Scribe overlay. We have developed the techniques in the context of Massively Multiplayer Online Games (MMOGs). While Scribe provides a possible platform for the scalable deployment of MMOGs, game developers may leverage the techniques to enhance the design of real-time interactions between players in the game world.

I. Introduction

In recent years, there has been much interest in peer-to-peer (P2P) multicast in response to the impracticality of the deployment of network-layer IP multicast. P2P multicast organizes end hosts (peers) into networks overlaid on the Internet that functions as a communication platform.

Scribe [1] is a P2P multicast design that is developed on top of Pastry [2], a structured P2P overlay. Scribe achieves scalability by leveraging the underlying Pastry substrate that adopts a decentralized P2P model through the use of Distributed Hash Table (DHT). As a result, Scribe is scalable with respect to supporting large numbers of multicast groups, with potentially large number of members in each group.

With respect to performance, Scribe adopts a light-weight technique called Proximity Neighbor Selection (PNS) to improve the latency associated with multicast communication. Traffic path in a P2P overlay is through overlay hops from one peer to another. PNS, in simple terms, selects peers that are topologically close to each other to form the traffic path.

We study Scribe in the context of Massively Multiplayer Online Games (MMOGs). MMOGs are traditionally deployed in a client-server (CS) model. CS model achieves scalability through partition techniques. A game world is partitioned into multiple zones, and each zone is processed by a zone server in a server cluster. A zone may be further partitioned into multiple *area of interest* (AOI), and a player only receives a subset of the *state update flow* that represents the latest states of the objects in his AOI. It should be noted that when a large number of players connected to a central server cluster, while the computation can be distributed, the possible congestion of bandwidth stress can not be avoided.

In response to the critical scalability issues of the CS model, distributed game design using a P2P architecture and, in particular, Scribe has started to emerge in the research field and includes [3] and [4]. The fundamental idea is to leverage Pastry properties such as self-organising, de-centralized control, fault-tolerance and reliability. Ideally, a P2P overlay such as Pastry can dynamically scale up and down with the number of peers, and remove the inherent scalability issues associated with the CS model.

In this paper, we argue that a standard Scribe setting is not optimal with respects to latency performance and bandwidth distribution. This is particularly true in the context of MMOGs, in which the quality of the games requires significant resources such as bandwidth and low latency level. Hence, the challenge is to further improve the performance of Scribe and, at the same time, maintain the integrity of Scribe as a structured P2P overlay.

To that end, we have developed a couple of techniques in this paper to enhance the multicast performance of Scribe in supporting MMOGs. The key to our techniques is the recognition that while the build-in PNS of Scribe can effectively reduce the latency associated with the initial overlay hops of a traffic path, the final hop of the traffic path is largely selected without any proximity consideration, and usually is a major contributor to the overall delay. As a result, demonstrated by the simulation results, both of our techniques can improve the latency performance for more than 30 percent, and the second technique further improves the bandwidth distribution by a factor of 2.7, comparing with what can be achieved in a standard Scribe setting.

The rest of the paper is organized as follows. In section II, we briefly sketch Pastry and Scribe. In section III, we reveal how bandwidth and latency impose constraints on the interaction metrics in MMOGs. In section IV, we briefly review a distributed game designs using Scribe to achieve deployment scalability. In section V, we present the techniques to improve the multicast performance of Scribe in supporting the game distributed design. We discuss related work in section VI and conclude in section VII.

II. Pastry and Scribe

The core task of Pastry is to route a message from the source to the destination within an expected $\log_{2^b} N$ (b is usually chosen to be 4, and N is the number of nodes in the overlay) hops. This is achieved through an ID prefix match scheme. For instance, if a message with the key 'd35' is to be routed from a source peer with ID '356', the overlay traffic path for the message could be 356 → d58 → d38. Note that each hop passes the message to a node whose ID shares one more prefix with the key than the previous node. The message arrives at the destination node whose ID is numerically closest to the message key.

The process for Scribe to build a multicast tree on top of Pastry is a straight forward process. Following the above example, if the same message is routed from a different source node whose ID is '894', the traffic path could be 894 → d58 →d38. Since both paths overlap at a common node 'd58', the multicast tree is then established with the reverse path forwarding scheme, and the node d58 serves as an interior node in the tree.

In order to route a message based on this prefix match scheme, each node in the overlay stores a routing table, a neighbour set and a leaf set.

III. Current Interaction Features of MMOGs

Kim J., et al. [5] conducted a traffic analysis on Lineage II, a leading 3D MMOG. In Lineage II, the average traffic generated by a client towards the server is 2kbps, mainly to transmit action commands, and the average traffic generated by the server towards per client is 10 times higher than the client generated traffic and at 20kbps, mainly transmitting the state update flow.

In [4], we conducted a traffic trace at the client side for World of WarCraft (WOW), which is arguably the most popular MMOG on the market at present. The traffic trace has shown that the bandwidth usage for the state update flow is approximately linearly correlated with the number of players in a typical AOI.

Based on the traffic trace in [4] and [5], it can be inferred that bandwidth is the key component that restricts the interaction metric with respect to the population level of an AOI.

The other interaction metric is the responsiveness to player actions. To tolerate the latency heterogeneity of the Internet, current MMOGs usually only support slow-paced actions that do not require quick responsiveness. Nevertheless, the quality of game playability with respect to action responsiveness could be significantly influenced by a high latency [6]. On the other hand, the latest development trend is to have fast-paced (quick responsiveness) actions integrated into MMOGs that traditionally only support slow-paced actions. For instance, BigWorld [7] is a server technology that has this design potential as one of its principal aims.

To conclude, current MMOGs have a small population of players in an AOI (e.g., in a number of tens) who can interact with each other concurrently, and slow action responsiveness. Those constraints are directly imposed by bandwidth shortage and high network latency.

IV. Distributed Game Design Using Scribe

In [4] , we presented a distributed game design by considering the advanced MMOG settings. The use of Scribe is justified by considering the significant asymmetry between the client generated traffic (action commands) and the server generated traffic (state update flow). It is proposed to use Scribe to multicast game state update flow in order to save bandwidth usage and promote the population level of an AOI. However, action commands can be unicast to the root of the multicast tree due to the low bandwidth usage incurred. In the design, the root of a multicast group serves as a server for an AOI. Multiple AOIs can form a huge game virtual world taking advantage of the scalability nature of the P2P overlay, and leveraging the partitioning techniques used in MMOGs.

The simulation conducted in [4] based on a Pastry overlay of 5000 nodes has shown that, to support a population level of 100 players in an AOI, a central server model would incur an upstream bandwidth usage of 5 Mbps (50kbps x 100 = 5 Mbps) at the server for the state update flow, whereas Scribe can build a tree with *the length less than four* to support the same level of population and only requires low end ADSL bandwidth capacity of peers.

V. Enhancing the Multicast Performance of Scribe

For the distributed game design discussed in the previous section, we can assert that Scribe scales well in terms of bandwidth distribution and latency performance *provided we are assured that a small number of hops is an indication of low latency*.

However, it is a general concern in the research field [8, 9, 10] that the uniform distribution of NodeID in Pastry could lead to long hop distance. In addition, large-sized Scribe multicast tree may have nodes subject to excessive stress. In this section, we first discuss the multicast performance of a global overlay. We then present techniques to enhance the multicast performance of Scribe. We present simulation procedures and results along the way in assisting the discussion.

A. Multicast performance of a global overlay

Proximity Neighbor Selection (PNS) is a light-weight technique used by Pasty to control the distance traveled by each hop along a traffic path. In simple terms, a Pastry node pings a number of candidate nodes in the overlay and selects one that is closest to itself as an entry for its routing table. As a result, the more candidate nodes exist, the more chance a node can find a closer entry for its routing table. However, the number of candidate nodes for a routing table decreases exponentially as the level of the routing table increases due to the design feature of prefix matching. This exponential reduction of candidate node size would result in gradual deterioration of the PNS effectiveness and increased hop distance along a traffic path.

In addition, while PNS may effectively reduce the distance of the initial hops of a traffic path, the final hop is usually routed towards an entry in a node's *leaf set* in a topologically randomized manner without the application of PNS. To demonstrate this, we established a global Pastry overlay of 10000 nodes using a GT-ITM Transit-Stub topology [11]. Figure 1 is the CDF of the final hop distance vs. the non-final hop distance.

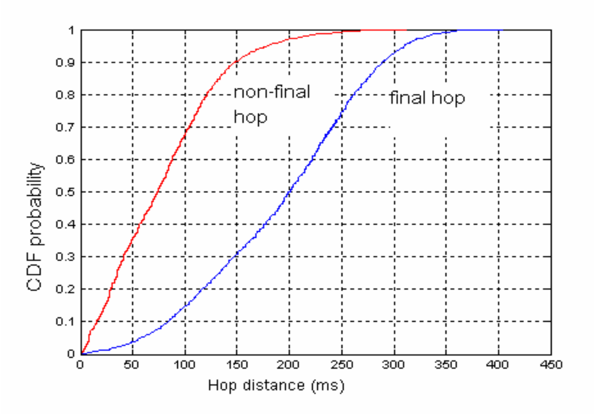

Figure 1: CDF of hop distance

It is shown in Figure 1 that the distance of the final hop dominates the total distance of a traffic path. For instance, given our topology, there are 10 percent of non-final hops compared with 70 percent of final hops that are experiencing a latency level of more than 150 ms.

We hence envisage that to improve the latency performance of Scribe to promote action responsiveness for games, the key may rely on further improving the final hop latency of the traffic path in Scribe.

B. Apply PNS to the final hop

1) Motivation and approach

Clustering technique is a widely discussed topic in the research field [8, 9, 10]. The idea is to find nodes that are close to each other, and provide this locality information to applications to achieve better performance. For instance, Ratnasamy et al. [10] proposed to use the landmark scheme to have nodes clustered into bins.

Assuming nodes can be clustered into bins, if an AOI server is located in a bin, the challenge is to find a short path between a peer in a different bin and the AOI server, while maintaining the integrity of Pastry overlay. We have discussed in the previous section that PNS is a light-weight technique in controlling distance of initial hops of Pastry traffic path, and identified that the final hop dominates the distance of the total traffic path because PNS does not apply to the final hop. With those in mind, we here propose an approach to apply PNS to the final hop as follows.

First, instead of establishing a global P2P overlay, through clustering techniques, multiple local overlays can be established for better use of resources. Moreover, a major concern over Pastry design is the possibility of undesired hops due to the random properties of the design. For instance, a path might traverse across bins before coming back to a nearby node in the local bin. By clustering nodes, this possibility is eliminated because the traffic path is constrained to local links. In this paper, we rely on available techniques [8, 9, 10] for node clustering. However, we envisage that game distributed design might have specific needs for clustering techniques, for which we will leave it to the future work to investigate.

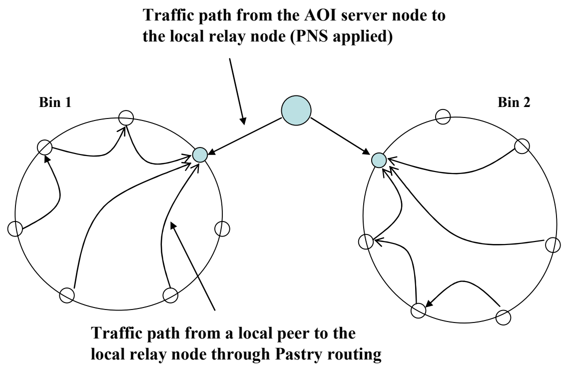

Figure 2: PNS applied to the final hop

Second, when peers in a local bin connect to an AOI server located in a different bin, instead of connecting to it through the global overlay routing, peers are routed to a local *relay node* that is close to the AOI server node as shown in Figure 2. Note that in doing that, the hop from the relay node to the AOI server node becomes the final hop of the traffic paths for peers.

We apply PNS to the choice of the relay nodes to improve the final hop latency performance. When selecting a relay node in a local overlay, an AOI server node can ping a number of nodes in that local overlay and pick the closest as the relay node.

Note that when two nodes are located in different domains, the final hop usually traverses a major link across domains. By applying PNS to the final hop, we are actually exploiting network proximity at major link level. Considering the final hop across domains usually incurs the longest distance traveled, the application of PNS to the final hop should be more effective in reducing the total distance traveled.

2) Simulation procedure and results

To test the effectiveness of the final hop PNS, using the Transit-Stub topology in 5.1, we developed a simulation procedure as follows. First, we take advantage of the locality properties of the Transit-Stub topology by clustering nodes based on which domain they belong to. Based on transit domains, we have three clusters of nodes. To achieve finer granularity, we further group nodes based on which stub domain they belong to. We are able to have 24 clusters of nodes because there are 24 stub domains in our topology model. For transit domain level clustering, we have 3334 end-hosts on average in each domain. For stub domain level clustering, we have 420 end-hosts on average in each domain. Our intention is to compare the performance of simulation between different levels of granularity in terms of topological awareness achieved. For instance, transit domain level clustering achieves less topological awareness than stub level clustering does. In other words, the average distance between a pair of nodes in a transit domain is higher than that in a stub domain.

Second, a local structured P2P overlay is established following the Pastry design. It should be noted that for a node to join a local overlay, the node only needs to contact a node that is already in the local overlay to be bootstrapped into the overlay network. Aside from this, the procedure of establishing a local overlay makes no difference from that of establishing a global overlay.

Finally, we pick randomly a pair of end-host nodes in the topology, one as the source node and the other as the destination nodes (a potential AOI server). We compare the distance traveled from the source to the destination between three settings: 1. A global overlay. 2. Three clustered overlays with final hop PNS applied. 3. Twenty-four clustered overlays with final hop PNS applied. The results are presented in Figure 3.

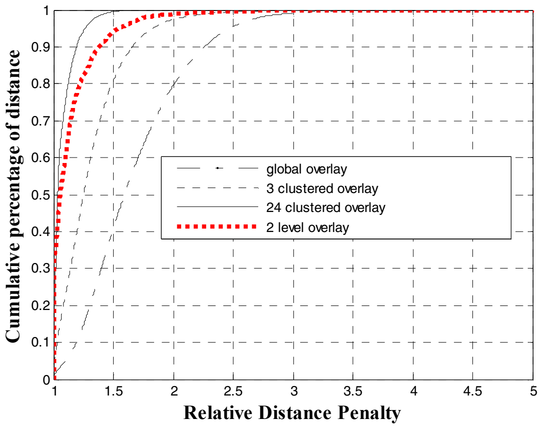

Figure 3: Cumulative distribution of RDP

Figure 3 compare Relative Distance Penalty [2] of the three settings. RDP is defined as the ratio between the routing distance and the underlying topology distance, and serves as an indication of the routing performance. Figure 3 shows that, while 10 percent of the routing distance has achieved RDP of 1.2 in the global overlay, approximately 30 percent of the routing distance in the three clustered overlay setting have achieved the same RDP level. The percentage of this RDP level has further increased to 90 percent in the twenty-four clustered overlay setting. The average RDP of the 24 clustered overlay setting is 1.1. In other words, it is nearly as good as the underling topology distance that is the shortest distance can be achieved. The curve labeled "two level overlay" will be discussed in the next section.

To conclude for the simulation results, the techniques of clustering and final hop PNS can significantly improve the latency performance of P2P multicast using Scribe. Game developers could take advantage of the techniques to promote action responsiveness for MMOGs.

C. Two-level overlay

1) Motivation and approach

We further consider the population level of an AOI that is dependent on the bandwidth usage. Recall that an AOI server node connects to peers in a local bin through a relay node according to the technique discussed in the previous section. Hence, an AOI server node would have a number of direct connections (number of children entries or fan-out) to relay nodes that is equal to the number of bins clustered in the worst case.

The inflexibility of bandwidth stress at the AOI server node could create a potential bottleneck depending on the desired population level of an AOI. For instance, 24 clustered overlays have achieved the best latency performance in our simulation but would result in a fan-out of 24 at the AOI server node, which is the root of the multicast tree for the state update flow. For instance, if an AOI server node is to support a population level of an AOI of 200 players, each direct connection of that server node would be subject to an upstream bandwidth stress of 100 kbps, projected from World of WarCraft traffic trace [4], and result in a total bandwidth stress of 2400 kbps at the server node. Such a stress could be expensive and discourages a client computer (a peer) to take the role of an AOI server.

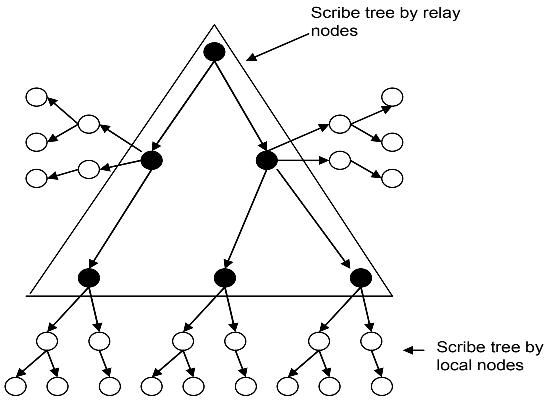

Figure 4: Two-level overlay

We hence propose to establish a second Pastry overlay among relay nodes that further distribute bandwidth stress. Instead of connecting to the server node directly, relay nodes self-organize into a second level of overlay and route to the server node through the Pastry routing scheme. Along the process, a Scribe multicast tree can be built on-the-fly to distribute the bandwidth stress accumulated at the server node. In addition, the second overlay can take advantage of the build-in Pastry PNS technique to exploit the network proximity at a higher level. In other words, while clustered overlay allows peers to exploit network proximity within the local bin at the local link level, the second level overlay allows relay nodes to exploit network proximity across the bins at the major link level. The two-level overlay is conceptually illustrated in Figure 4.

2) Simulation procedure and results

Following the establishment of 24 clustered overlays, we randomly select one end-host that is connected to a transit node in the topology as an AOI server node. The server node then pings a number of nodes in a local overlay to select the closest node as the relay node for that overlay. Note that the sever node only needs to know one node in any local overlay to be able to obtain the routing table of that node and has access to a list of nodes in that local overlay for pinging purpose. After the choice of relay nodes for each clustered overlay has been made, the relay nodes and the AOI server node then self-organize into a second level Pastry overlay. When a peer joins the game session hosted by the AOI server node, it routes to the local relay node to join the local Scribe tree, and then the second level Scribe tree can be established when relay nodes route to the AOI server. The reverse path forwarding scheme is used for state update traffic in the trees established (Figure 4).

To test the performance of the two level overlay, we experimented with a population level of an AOI of 240 players, which is a high population level for an AOI considering most current MMOGs only has a small AOI population. We have distributed those players uniformly among the local overlays. We measure the bandwidth stress of the AOI server node and relay nodes for those nodes being subject to most bandwidth stress as roots of the multicast trees. The results are shown in Figure 5.

Figure 5 shows that the two-level overlay has achieved an average fan-out of 8.76. If each connection for a child needs an approximately 100kbps of upstream bandwidth stress projected from WOW traffic, the two-level overlay can effectively constrain the bandwidth stress of root nodes below 1Mbps (8.76 x 100kbps < 1000kbps), which is feasible for current high-end ADSL links. In terms of latency performance, Figure 3 shows that the CDF distribution of RDP of the two-level overlay is slightly worse than the setting of the twenty-four clustered overlays.

In table 1, we summarize the comparison between the four settings in supporting the population level of an AOI of 240 players: 1. the global overlay. 2. The two-level overlay. 3. The 24 clustered overlays. 4. The central server model.

Table 1 shows that, the cost to the central server would be prohibitively high at 24Mbps. In comparison with the central server model, the twenty-four clustered overlay setting and the two-level overlay setting have improved the bandwidth distribution by factors of 10 and 27 respectively. The costs for both settings are the increases of latency of 11% and 17% respectively, which are much less than the cost of the standard Scribe setting of 67% latency increase.

Furthermore, both the settings of the two-level overlay and the twenty-four clustered overlays have improved the latency performance up to 30 and 34 percent in mean value respectively over the standard Scribe. We believe such an improvement is significant to the action responsiveness. For instance, highly real-time sensitive actions such as shooting require the latency to be less than 150ms [6]. If the IP routing distance from a player to the server is 100ms, the use of standard Scribe would increase the latency to 167ms that could have severe negative influence on the game playability. However, the use of our techniques only increases the latency to 111ms and 117ms, which might not be perceived by the users.

To conclude, we have demonstrated that our techniques can push the AOI population to a higher level, which would otherwise incur a higher cost to a standard Scribe setting with respect to latency, or an even worse prohibitive cost to a central server with respect to bandwidth usage.

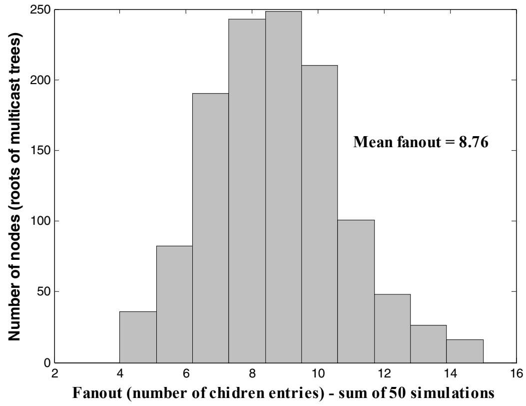

Figure 5: Fan-out of roots of multicast trees - Two level overlay – population of AOI = 240 players

Table 1: Comparison between four settings

	Global overlay	Two level overlay	24 clustered overlays	Central server model
Mean RDP	1.67	1.17	1.11	1
Expected Fan-out at roots (AOI servers)	23	8.76	24	240
Expected bandwidth stress for MMOG traffic	2.3Mbps	876kbps	2.4Mbps	24 Mbps

VI. Related Work

P2P systems can be generally classified into two kinds: *Unstructured* and *Structured*. The latter achieved significant performance improvements over the former in terms of load balancing and efficient message routing, through the use of Distributed Hash Table (DHT) [12]. Other popular structured P2P overlay designs include CAN [13], Chord [14] and Tapestry [15], each provides a scalable platform for P2P multicast. Castro M. et al. [16] conducted a head-to-head comparison of P2P multicast between CAN and Scribe. The general findings are that, at the similar management and maintenance cost, Scribe outperforms CAN multicast on the order of 20% - 50% with respect to latency performance.

In comparison with Chord and Tapestry, the main advantage of Pastry is its less complexity. As a result, Scribe requires less management and maintenance overhead that benefits the scalability of P2P multicast. For instance, Bayeux [17] is built on top of Tapestry and supports multiple multicast groups. A root of a multicast group in Bayeux has to keep a list of all group members and all management traffic has to go through the root, whereas in Scribe the group member information is distributed over the nodes.

While we have based our work on Scribe for reasons briefly discussed above, we believe that the techniques we developed can also be applied to CAN, Chord and Tapestry since they are all based on Distributed Hash Table.

The efforts to enhance the point-to-point routing performance of structured P2P overlay include Brocade [18] and Expressway [19]. Brocade separates "super-nodes" from normal nodes, and establishes a second overlay among the super-nodes. It is assumed that each node knows the AS domain that it belongs to (e.g., using BGP report). Each super-node maintains a "cover set" of a list of nodes in his AS domain and exchanges this information with other super-nodes for the selection of paths across domains. In selecting the path across domains, Brocade relies on the exchange of state information of cover sets among super-nodes, which could have scalability concerns when the cover set involves a large number of nodes. To address the scalability concerns of Brocade, Expressway uses a summarization scheme to advertise the cover set.

The major advantage of our technique in comparison with Barocade and Expresseway is its light-weight nature by only applying PNS at the final hop of Scribe traffic path, whereas both Brocade and Expressway need to run different protocols and algorithms to maintain the super-overlay, reducing the scalability. Furthermore, the use of the second overlay in Barocade and Expressway is solely for the purpose of enhancing the point-to- point latency performance whereas our work has used the second overlay to distribute bandwidth stress, in addition to enhancing latency performance.

To our best knowledge, we are the first to try to enhance the P2P multicast performance of Scribe in the context of MMOGs.

VII. Conclusions

In this paper, we have developed a couple of techniques to enhance the P2P multicast performance of Scribe with respects to latency performance and bandwidth distribution.

The key to the first technique is to identify that the final hop of Scribe traffic path is largely topologically randomized and incurs the longest distance traveled by a message. The technique hence leverages clustering techniques and applies Proximity Neighbor Selection (PNS) to the final hop of the multicast traffic path. Simulation results have shown that the technique can improve the latency performance for more than 30 percent in comparison with what can be achieved in a standard Scribe setting.

The second technique builds a two-level overlay: the higher level among nodes located close to network access points, the lower level among nodes in a local domain. While PNS can be applied at both levels for latency performance, the bandwidth stress required by applications can now be distributed among the nodes in the higher level overlay. The simulation results have shown that, while latency performance is slightly worse than what is achieved by the first technique, the second technique has improved bandwidth distribution up to 2.7 times, in comparison with what can be achieved by a standard Scribe overlay.

The techniques we proposed could benefit the distributed game design over the structured P2P system in promoting the interaction metrics for better game playability. In considering actual traffic required by advanced MMOGs, we have demonstrated that the techniques can push the interaction metrics to a higher level, which would otherwise incur a higher cost to a standard Scribe setting with respect to latency, or an even worse prohibitive cost to a central server with respect to bandwidth usage.

Acknowledgment

This work is supported by the Co-operative Research Centre for Smart Internet CRC (SITCRC) and the University of Wollongong (UOW), Australia.

References

[1] M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron, "SCRIBE: A large-scale and decentralized application-level multicast infrastructure," *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, vol. 20, 2002.

[2] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," presented at Proc. IFIP/ACMMiddleware, Heidelberg, Germany, 2001.

[3] B. Knutsson, H. Lu, W. Xu, and B. Hopkins, "Peer-to-peer support for massively multiplayer games," presented at INFOCOM 2004, 2004.

[4] X. Jiang, F. Safaei, and P. Boustead, "A Performance Evaluation of Structured Peer-to-Peer Network in Supporting Massively Multiplayer Online Role Playing Games," presented at International Symposium on Performance Evaluation of Computer and Telecommunication Systems, Calgary, Canada, 2006.

[5] J. Kim, D. Chang, and T. Kwon, "Traffic Characteristics of a Massively Multi-player Online Role Playing Game," presented at Netgames'05, New York, USA, 2005.

[6] X. Jiang, F. Safaei, and P. Boustead, "Latency and Scalability: A Survey of Issues and Techniques for Supporting Networked Games," presented at International Conference on Networks (ICON 2005), Kuala Lumpur, Malaysia, 2005.

[7] Bigword Technology, www.bigworldtech.com., accessed on 12/4/2006.

[8] X. Y. Zhang, Q. Zhang, Z. Zhang, G. Song, and W. Zhu, "A Construction of Locality-Aware Overlay Network: mOverlay and Its Performance," *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, vol. 22, pp. 18-28, 2004.

[9] R. A. Ferreira, S. Jagannathan, and A. Grama, "Enhancing Locality in Structured Peer-to-Peer Networks," presented at Tenth International Conference on Parallel and Distributed Systems (ICPADS?4), 2004.

[10] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Topologically-Aware overlay construction and server selection," presented at INFOCOM, 2002.

[11] E. Zegura, K. Calvert and S. Bhattacharjee., "How to Model an Internet," presented at IEEE INFOCOM, 1996.

[12] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *Communications surveys and tutorials, IEEE,* vol. 7, pp. 72-93, 2005.

[13] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Application-level multicast using content-addressable networks," presented at Proceedings of NGC, Nov. 2001.

[14] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," presented at Proceedings of SIGCOMM 2001, 2001.

[15] B. Zhao, J. Kubiatowicz, and A. Joseph, "Tapestry: An infrastructure for fault-resilient wide-area location and routing," U.C.Berkeley, Tech.Rep. UCB//CSD-01-1141, April 2001. April 2001.

[16] M. Castro, M. B. Jones, A.-M. Kermarrec, A. Rowstron, M. Theimer, H. Wang, and A. Wolman, "An Evaluation of Scalable Application-level Multicast Built Using Peer-to-peer Overlays," presented at INFOCOM, 2003.

[17] Z. S., B. Zhao, A. Joseph, R. Katz, and J. Kubiatowicz, "Bayeux: An Architecture for Scalable and Fault-tolerant Wide-Area Data Dissemination," presented at NOSSDAV, June 2001.

[18] B. Y. Zhao, Y. Duan, L. Huang, A. Joseph, and J. Kubiatowicz, "Brocade: Landmark Routing on Overlay Networks," presented at 1st International Workshop on Peer-to-Peer Systems (IPTPS02), 2002.

[19] Z. Xu, M. Mahalingam, and M. Karlsson, "Turning Heterogeneity into an Advantage in Overlay Routing," presented at INFOCOM, 2003.