An Efficient Algorithm for Selection and Management of Island Multicast

Abbas Bradai, Toufik Ahmed
CNRS-LaBRI University of Bordeaux-1
351, Cours de la libération
Talence, 33405
{bradai, tad} @labri.fr

***Abstract*—Although IP multicast techniques were proposed a long time ago and despite of their advantages, they are still not widely deployed due to the absence of multicast support in some routers/domains and inter-domain management issues. On the other hand, in the most of recent internet applications, where the average consumed bandwidth is measured by hundreds of Kbits per second and where the support of large-scale distribution is important, the IP multicast becomes more than a necessity. In this paper, we propose a new approach for extending the scope of IP multicast in overlay applications. We selected some overlay nodes to be used as fan-out multicast nodes and then created an IP multicast islands around each fan-out node. These multicast islands are connected with each other using unicast overlay links. This selection of fan-out nodes is based on a distributed version of K-means algorithm and GNP (Global Network Positioning), a distributed technique to measure the distance between nodes. We further propose a preventive fault tolerance mechanism for packet loss across islands. Finally, the simulation results verify the optimality of our approach in terms of link stress minimization and end-to-end delay reduction.**

Keywords-Application-level; multicast; IP multicast; Island multicast; clustering; k-means

I. INTRODUCTION

IP multicast is considered as an efficient mechanism to deliver large-scale content over internet, especially for video streaming, to save bandwidth consumption and reduce end-to-end delay. Since it is impossible to provide a global multicast solution at the network level, application level multicast (ALM) has been proposed as an alternative. In ALM, some group members form an overlay network and content is distributed via unicast by relaying packets from one node to another. Peer to peer (P2P) network is one of the example of these overlay networks.

Two main types of architectures are generally considered to achieve the ALM in P2P networks. In tree-based overlay [1], nodes are organized as a single, or multiple trees that connect the source of the content to clients. The flow of content follows a logical order in which content flows from a parent to its children nodes in the form of a tree. This type of overlay is easy to implement and to maintain, and it minimizes the end-to-end delay. However it has few limitations: (1) Upload bandwidth is limited by the minimum upload bandwidth of the intermediate nodes in the path. (2) Higher instability due to the frequent arrival/departure of the nodes (churn). On the other hand, in mesh-based overlay [2], nodes are connected to each other in a directed mesh to achieve content delivery from source to destination. This architecture guarantees a low cost, simplicity of structural maintenance and strong resilience to nodes failure or departure.

Recent research investigations have focused on overlay graph optimization and efficient routing protocols' design, to minimize the transmission delay for real-time applications [3] [4]. Most of the proposed solutions hide the underlying physical topology and do not take advantage of the local multicast capacity of the network. In [5], authors proposed a hybrid technique, combining ALM with IP multicast, called Island Multicast (IM), where multicast-capable domains called Islands are interconnected using overlay unicast connections to extend the scope of IP multicast by achieving global multicast solution. Nodes relaying between islands are called bridge nodes (BN). Extensive researches have been carried on this new approach, to optimize the bridge node selection and loss recovery [6]. In [7], authors propose a formal method to assign nodes to islands based on fuzzy recognition. It is observed that most of research on IM does not consider the construction of Islands. They try to optimize the interconnection of islands to achieve a global connectivity by optimizing new elements assignment or enhancing existing routing protocols.

Our work differs from the other research works by considering the problem of overlay applications' optimization by introducing regions of multicast or "islands". We propose a mechanism for supporting hybrid multicast, in which selected overlay nodes are turned into fan-out multicast nodes to enable the delivery of IP multicast content within the Autonomous System (AS). We design a partitioning model based on the distributed version of K-means algorithm and a distributed approach to measure the distance between overlay nodes (GNP) [8]. K-means allows an optimal partitioning of overlay nodes into *k* islands in such a way that it maximizes the inter-island distance and it minimizes the distance between the nodes of the same island.

Several distance measurement techniques have been proposed such as IDMaps [8] and the triangulated heuristic.

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) in the ENVISION project, grant agreement 248565.

GNP is still the most scalable, the less costly and the more accurate. Indeed the IDMaps requires O (n^2) messages to compute hosts coordinates instead of O ($n*d$) in the case of GNP, where n represent the number of hosts in the network and d it dimensionality. The IDMaps requires some special hops servers to maintain a virtual topology map of the Internet, while any ordinary host may be a GNP landmark. Additionally GNP achieves very good accuracy and performs far better than the triangulated heuristic.

The rest of the paper is organized as follows. In Section II, we describe the IM creation mechanism. In section III, we describe in detail the proposed mechanism namely Efficient Island Multicast (EFIM) integrating the basic mechanism of overlay construction, the join/leave mechanism and the fault recovery techniques. In Section IV, we present some illustrative simulation results. The conclusion and future work are presented in V.

II. DESIGN OF IM CREATION

A. Problem formulation

To design an IM architecture, the first step is to aggregate some nodes together to form islands, then elect a local leader for each island which will act as a source of multicast into the island. A new node joining the system will be inserted to one of these islands depending on some parameters such as its distance from the island/island's leader, its network accessing frequencies, etc. In Figure 1, we show an example of a partitioned overlay network into three islands having a leader for each island. In the proposed mechanism, we are trying to resolve following issues: (1) How to partition efficiently nodes into islands? (2) Which nodes are selected to act as fan-out-multicast nodes? (3) Which partitioning parameters to be considered and how to measure them efficiently in order to keep the scalability of the partitioning step? (4) How to affect a new node arriving into an existing island? And (5) The mechanism of overlay construction, join/leave mechanism and fault recovery techniques in EFIM?

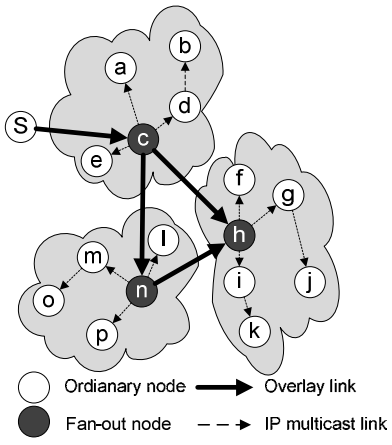

Figure 1. Island multicast overview

B. Creating Islands

We consider an already built mesh-based overlay and we want to activate the IM mode once some parameters are satisfied. For example when link stress (number of copies of the same packet transmitted over a physical link) exceeds a critical value, and/or when latency becomes very important. These parameters will be discussed in details in section IV.

Let suppose the overlay network composed of n nodes with d dimensions to be partitioned into S islands. Let x_i (i=1, 2, … , n) be the property vector of d dimension, that denotes the i^{th} node of the overlay network. Let x_{ij} be the j^{th} property value of x_i where i = {1, 2, … , n} and j = {1,2, … , d}, and let *distance* (x_k ,x_l) be the multidimensional distance between the nodes specified by the property vectors x_k and x_l respectively. Our approach for creating islands is based on two well-known algorithms in the statistic and distributed systems: K-means and Sound/echo algorithms, and on a distributed approach to measure the distance between nodes which is based on the Global Network Positioning (GNP) approach. We will explain this approach in the following sub-section.

1. K-means clustering algorithm

K-means is considered as the most successful common used method for partitioning data into clusters. K-means algorithm is simple to implement and computationally attractive due to its linear time complexity because it is not based on computing the distances between all pairs of data points. K-means is a non-hierarchical partitioning approach. Its inputs parameters are a number k of clusters to which the data will be partitioned to, a set of n-dimensional data entities d_i, and an initial m cluster centroids. The algorithm unfolds in several iterations, in each iteration, each d_i is assigned to the nearest centroid c_j, then the new centroid of each cluster is recalculated. This operation is repeated until the termination condition is satisfied. One of the most widely used conditions, is that the algorithm converges if the sum of all mean square errors (MSE) within each cluster does no longer decrease from iteration to iteration. The number of iterations may be also fixed according to the algorithm application field. Note that the distance function, *distance* (d_i, c_j) defined by Eq.1 can also be chosen according to the algorithm application field, the Minkowski distance is the common metric widely used.

$$distance(d_i, c_j) = \sqrt[m]{\sum_l | d_{i,l} - c_{i,l} |^m} \qquad \text{Eq. 1}$$

Where $m \in N$ can be set to define a specific metric, for example the Euclidian distance (m=2) was taken into consideration in our experiments. Many methods were proposed to generate the initial cluster centroids, ranging from random centroids to complex ones that aim to achieve fast convergence of the algorithm. Other works focus on the convergence speed of the algorithm based on some geometric properties such as the trigonometric inequality [9], or the estimation of the number of clusters [10].

Basic K-means algorithm

1) *Generate a random k group centroids*
2) *Assign each object to the group having the closest centroids*
3) *When all objects have been assigned, recalculates the centroids of the K groups.*
4) *Repeat steps 2 and 3 until the considered termination condition will be satisfied: the centroids no longer move.*

---

2. Sound/Echo mechanism

In our solution, based on a distributed version of K-means, we need, as we note in the next section, to propagate some parameters in the network. For this purpose, we used Sound/Echo mechanism. The sound/echo mechanism [11] is an algorithm to distribute a piece of information across a general graph. It aims to reduce the number of sent messages and processing steps.

The algorithm is initialized by a node called *initiator*. This node later marks itself both as *engaged* and *initiator* and sends a SOUND message to all it neighbors. A node receiving SOUND message for the first time, marks itself *engaged* and sends a SOUND message to all it neighbors excepting the source of that SOUND message. A node marked *engaged* marks itself *not engaged* when it receives an ECHO message from all its neighbors, and sends an ECHO message to the node from which it received SOUND message for the first time. Finally, the algorithm stops when the *initiator* receives ECHO message from all it neighbors. This algorithm ensures that all nodes of the graph receive a message sent by an initiator, with the least cost in terms of exchanged messages and number of processing steps.

3. A distributed algorithm to create Islands

Based on the two algorithms presented above, we have developed an algorithm for an efficient partitioning of overlay nodes into multicast capable sub domains.

The initiator of the algorithm chooses randomly a set of *k* initial centroids, and spreads the list to all its neighbors with a SOUND message. Every peer receiving the list of centroids selects the nearest one and notifies the selected centroid. A node receiving SOUND message marks itself *engaged* and resend SOUND message to all it neighbors except the one from which it received the message (its predecessor). When a node and all of its neighbors have selected the nearest centroid (except the predecessor), it sends an ECHO message to its predecessor, and marks itself as *not engaged*.

When the *initiator* of the first SOUND message receives ECHO message from all its neighbors, (in other word all the network is partitioned to *k* clusters), it asks the centroids to compute the new centroid of each cluster. Therefore, a complete iteration of our K-means inspired algorithm is completed and terminates. Typically, the partitioning process requires more than one iteration. Thus, based on the adopted termination condition, the node initiator of the iteration decides if the clustering has to be enhanced by a new iteration.

4. Distance measurement

In the algorithm presented below, every node in the overlay network is assigned to the closest centroid. In order to measure the distance (similarity) between a peer d_i and a centroid c_j, we have adopted the Euclidian distance. Thus, the distance between the i^{th} peer and the j^{th} centroid is determined by the following formula (Eq. 2):

$$distance_{Euclid}(d_i, c_j) = \sqrt{\sum_l | d_{i,l} - c_{j,l} |^2} \qquad \text{Eq. 2}$$

Where $d_{i,l}$ and $c_{j,l}$ denote the l^{th} property value of the vectors d_i and c_j representing the i^{th} node and the j^{th} centroid respectively. Initially, we adopt the physical distance as selection metric, for node assignment to a particular centroid. In this case the property vector x_i is reduced to the physical location coordinates.

Round Trip Time (RTT) is widely used to predict and estimate host location in a network. This method allows having a good approximation of distance without using third party services as in the global positioning system (GPS) based on satellite signals, or GeoPing which use geographical IP mapping services. The non-scalability still remains the major problem of RTT. Indeed, to have a global view about the distance between all *n* nodes in the network, all these nodes have to ping each other. As a consequence, the complexity of this method is O (n^2). In our case we used GNP, based on absolute coordinates computed by modeling the network as a geometric space [8]. So every host in the network is represented by a point in the space. This technique is achieved in two steps. In the first step a small set of hosts named landmarks compute their coordinates in a chosen geometric space S. First they measure the inter-landmark round trip time using ICMP ping messages. We denote by $d_{li,lj}$ the distance between landmarks l_i and l_j.

In GNP, we aim to find a set of coordinates C_{l1}, … , C_{ln} for the *n* landmarks in a way to minimize the overall error between the measured distances and the computed ones. In other word, we seek to minimize the objective function:

$$f(C_{l1},...,C_{l,n}) = \sum_{l_i,l_j \in \{l_1,...,l_n\}} \varepsilon(d_{l_i,l_j}, d'_{l_i,l_j}) \qquad \text{Eq. 3}$$

Where ε is an error function, which is in our case the squared error:

$$\varepsilon(d_{l_i,l_j}, d'_{l_i,l_j}) = (d_{l_i,l_j}, d'_{l_i,l_j})^2 \qquad \text{Eq. 4}$$

With this formulation, coordinates computation is reduced to a multi-dimensional global minimization problem, which can be approximately, resolved using one available method such as the Simplex Downhill method [12]. Once the coordinates of the landmarks C_{l1}, … , C_{ln} are computed, they are disseminated in the network.

In the second step, based on the landmarks coordinates, every node *H* in the network values its coordinates regarding the same geometric space. Thus, the node measures its RTT to the N landmarks using ICMP ping messages. Let $d_{H,li}$ be the measured distance between the node H and the i^{th} landmark. Then H computes its own coordinates C_H in a way to minimize the overall error between the measured distances and the computed ones. Formally the objective function to minimize is:

$$f(C_H) = \sum_{l_i \in \{l_1,\ldots,l_n\}} \varepsilon(d_{H,l_i}, d'_{H,l_i}) \qquad \text{Eq. 5}$$

Where ε is an error function, which is always the squared error. Note that for a *d* dimensional geometric space we must use at least *d*+1 landmark nodes.

III. ISLAND MULTICAST MANAGEMENT AND LOSS RECOVERY

In this section, we describe in detail the basic mechanism of overlay construction, join/leave mechanism and fault recovery techniques in EFIM.

A. Overlay construction

The ultimate goal of our protocol is to take advantages of IP multicast, especially in terms of end-to-end delay reduction and link stress minimization. For these reasons, the overlay construction and the IM management should be simple with low setup and minimum overhead.

The overlay can be modeled by a directed graph G (P, E) where P is the set of nodes and E is the set of edges. The pair of nodes (p, q) ∈ E if *p* deliver the stream to *q* whereas *q* is an out-neighbor of *p* and *p* is an in-neighbor of *q*. We denote by T(*p*) the set of its out-neighbors and I(*p*) the set of its in-neighbors. The *Card* (T(*p*)) is the output degree of p and *Card* (I(*p*)) is its input degree.

In our algorithm, a new host H broadcasts first a join message to obtain a list of hosts in the system. It pings these hosts and selects the nearest *k* nodes. Then, it pings the neighbors of these selected nodes, and selects *k* closest ones from all the hosts (the *k* initial hosts and their neighbors). The new host H repeats this process until the round-trip time is lower than a certain threshold, or the number of iterations exceeds a certain value. At the end of the process, the new host selects from its current *k* closest hosts at most *l* nodes with highest forwarding bandwidth as it parents, where *l* is it input degree, proportionally related to it download capacity.

Figure 2 illustrates the overlay construction in our scheme. Suppose *k*=5 and a node H, whose input degree is 2, wants to join the system. H first broadcasts a join message in the network. Let C, N, M, K and L, are the nodes that respond to this join message. Then the host H pings the direct neighbors of these nodes which are: B, J, and I and select the *k*=5 closest nodes from the all {C, N, M, K, L, B, J, I}. Let assume that Node H choose N, M, K, B and J. The same process is repeated with these five nodes until a termination conditions is satisfied. Since the considered number of nodes is limited to *k* in each iteration, the communication overhead for joining the overlay is low.

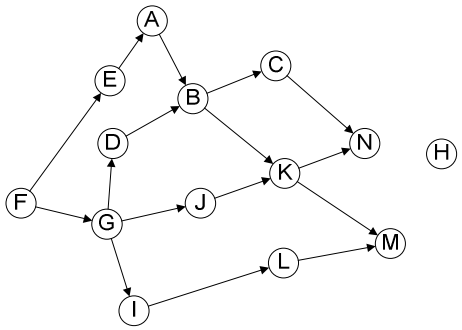

Figure 2. Overlay construction

B. Island detection

Once the overlay is constructed, and certain conditions for switching to IM mode are fulfilled (such as consumed bandwidth, the end-to-end delay and links stress), the supervisor of the AS decides to switch to the IM mode, and initiates the islands creation algorithm (cf. section II). The centroid of each island will be its unique ingress host, named the island's leader. It will be responsible for receiving data from outside the island and multicasting it into the island. An overlay structure is built to relay the different islands centroids to the source of the stream using the same overlay creation protocol described above.

In EFIM, after the IM mode is activated, a new host H first broadcasts a joining message request: "*join_mess"* to join the system. A member of the system responds by a "*resp_join_mess"* message along with the list of landmarks and the list of islands leaders described by their coordinates. The host H pings all the landmarks and calculates its coordinates in the adopted geometric space. Meanwhile, it computes it distance from each of islands leader. Then, it selects the nearest one c_i and sends to it an *island_join* message. The leader c_i adds the new host H to the island member list:"*Island_members*" and sends the data group and the control group address of the island to the new member. The first is used for multicasting the stream data while the latter is for control messages.

C. Island join operation

When a new host joins the session, and in addition to the two Multicast addresses, it receives the list of the currents island hosts *Island_members* arranged by their distance from the center of the island. This list is updated every time a node joins the island or leaves it.

The new island's member joins the multicast data group and starts receiving the data from the island leader in multicast mode, and stop receiving from it overlay parent(s). It also joins the control group and receives periodically a *Heartbeat* message from the island leader to detect any leader change, departure or failure. The others control messages are discussed in the fault tolerance mechanism part.

D. *Fault tolerance mechanism*

Two major issues can arise in IM: (1) island's leader failure and (2) packets loss during their transit across islands.

In our approach, we always try to keep the island's leader to be the nearest possible to the island center. This is the reason why we maintain and arrange the island members list, and communicate it to the island's members. Based on this, we hence propose a novel recovery scheme.

In the case of packets loss across islands, a natural way is to ask a retransmission from the upstream host. However such technique suffers from the problem of error correlation and implosion [13]. In the beginning, a retransmission will be asked to the upstream host. When the number of retransmission exceeded several times, a recovery node in the source island takes over. Thus, a new overlay relaying islands is constructed where the defective upstream node is replaced by it successor in the island members list as described above. We assume that the data source node marks each packet with an increasing sequence number. The island's ingress node detects errors by checking the sequence number. Whenever an error is detected, and the maximum number of retransmission attempts is exceeded, the ingress island's leader informs the upstream source node. This later asks its successor to take over. The new leader, adopting the same children and connected to the same parents, connects to the inter-islands overlay and starts to stream on the data multicast address, while the old leader disconnects from the overlay and starts receiving the data stream in multicast mode.

Two others leader's substitution scenarios are considered:

a) The island's leader leaves or crashes: In this case, the second member in the island host's list takes the responsibility to be a new leader. It connects to the inter-Islands overlay, based on the old leader children and parents. It sends a Heartbeat message, in the Control IP address, to all island's members which updates the member's list.

b) The new host joined the island has a lower distance from the center of the island than the current leader by a certain threshold. In this case the new host replaces the old leader in the overlay, sends a Heartbeat message to the multicast control address and starts transmitting data by the multicast data address.

IV. ILLUSTRATIVE SIMULATION RESULTS

To evaluate our scheme we have performed our simulations on internet like topologies, under the large used network simulator NS2 [14]. We generate 10 transits stub topologies with BRITE [15]. Stub domains are all connected to the backbone. In our simulations, each topology has 8 transit domains and 128 stub domains, consisting of 500 routers and about 3000 links. A group of hosts are randomly put into the network. A host is connected to a stub router with 1ms delay, while the delay of core links is randomly assigned. From the stub domains that consist of at least one host, we randomly select some nodes and set them to be multicast-capable. In our scheme, each new host obtains number of (at most 10) randomly selected hosts while joining. In our simulations, unless otherwise stated, the parameter S of our partitioning algorithm is set to 10, i.e. a stub domain is partitioned to 10 islands.

We also implement another tree-based ALM protocol for comparison, i.e., NARADA[16], one of the pioneering ALM protocols and its performance can serve as benchmark. We evaluate two important metrics in ALM, i.e., link stress and relative delay penalty (RDP). Link stress is defined as the number of copies of a packet transmitted over a certain physical link, and RDP is the ratio of the overlay latency from the source to a host to the delay along the shortest unicast path.

EFIM is compared with NARADA at different group size ranging from 16 to 2048. The link stress and RDP are represented in Figure-3 and Figure-4 respectively.

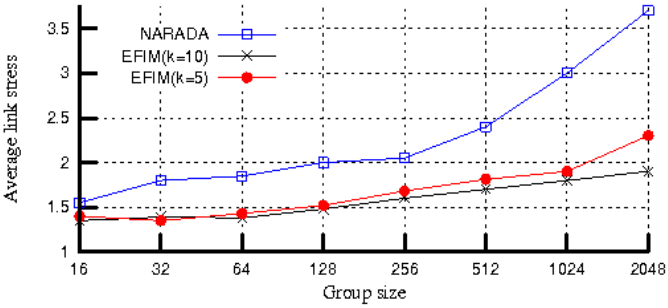

Figure 3. Link stress Vs group size

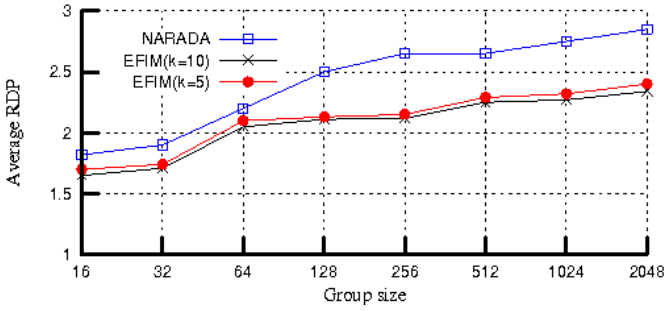

Figure 4. RDP Vs group size

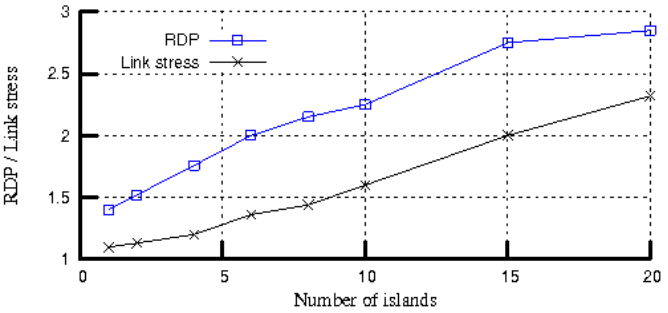

Figure 5. Link stress/RDP Vs number of islands

Figure-3 shows the average stress against group size. It is observed that the average stress increases with increasing group size; this is due to the large copies of a message

circulating in the network. Nevertheless, our scheme have 15-45% lower stress compared to NARADA due to the fact it selects the appropriate parents to hosts with the high upload bandwidth, and makes use of IP multicast. If we construct the overlay by selecting best parents from a largest set of nodes in each step (k=10 instead of k=5), link stress slightly decreases. This is because the probability of parents with highest bandwidth increases, accordingly minimizing the link stress.

Figure-4 compares RDP of our scheme with NARADA. It's observed that NARADA achieves small RDP because it tries to minimize the end-to-end delay. EFIM has a smaller end-to-end delay than NARADA especially when the group size is large. End-to-end delay approximately remains the same for different values of K because in each step we select parents with best upload bandwidth without considering the E2E delay.

Figure-5 shows the link stress and RDP of our scheme for different numbers of Islands (S) to which the overlay is partitioned. The group size is 512 and k=5. As expected, both the link stress and RDP decrease as the number of islands decrease. Naturally because this limits the numbers of overlay unicast link in favor of IP multicast links. The optimum is reached when we have one island per stub domain. Note that in this case the link stress and RDP are not equal to 1 as in pure IP multicast because the transit domains are not multicast capable.

V. CONCLUSION

The Internet today consists of IP multicast-capable Islands and IP multicast-incapable domains interconnected by IP multicast-incapable routers. Traditional ALM protocols only make use of unicast connections to form delivery trees and have not fully taken advantage of the local IP multicast capabilities. In this paper, we propose a fully distributed scheme for media streaming combining IP-multicast with ALM. Hosts are distributed and efficiently partitioned into islands having multicast features while the islands are interconnected by unicast connections. Simulations results show that our approach achieves low end-to-end delay and less link stress. For the future perspective, we aim to perform real test-bed evaluation for the more personalized services delivery over P2P network.

REFERENCES

[1] D.C. Arnold, G.D. Pack, and B.P. Miller, "Tree-Based Overlay Networks for Scalable Applications," 11th International Workshop on High-Level Parallel Programming Models and Supportive Environments (HIPS 2006). Rhodes, Greece, 2006.

[2] Damiano Carra, Renato Lo Cigno, and Ernst W. Biersack, "Graph Based Analysis of Mesh Overlay Streaming Systems," IEEE Journal on Selected Areas in Communications – JSAC, 25:1667–1677, December 2007.

[3] M. Mushtaq and T. Ahmed, "Hybrid Overlay Networks Management for Real-Time Multimedia Streaming over P2P Networks" in proceeding of Lecture Notes in Computer Science (LNCS), "Management of Multimedia Networks and Services", San José, California, USA, Vol. 4787, pp.1–13, 2007.

[4] U. Abbasi and T. Ahmed, "SWOR: An Architecture for P2P Scalable Video Streaming Using Small World Overlay," in proceeding of IEEE CCNC, Las Vegas 2010.

[5] K.. Roger Cheuk, S. Gary Chan, and J. Lee, Island Multicas, "The Combination of IP Multicast with Application-Level Multicast," IEEE International Conference on Communications, June 2004

[6] W.-P. Yiu, K.-F. Wong, and S.-H. Chan, "Bridge-node selection and loss recovery in island multicast," in Proc. IEEE Int. Conf. communications (ICC'05), May 2005, pp. 1304–1308.

[7] D. Cheng and J. Qian , "A Multicast Island Portioning Model of OMN Based on Fuzzy Recognition," IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.3, March 2007

[8] T.S. Eugene, and H. Zhang, "Predicting Internet Network Distance with Coordinates-based Approaches," IEEE INFOCOM, 2002

[9] C. Elkan, "Using the Triangle Inequality to Accelerate k-Means," Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003

[10] D. Pelleg and A. Moore, "X-means: extending K-means with efficient estimation of the number of clusters," Proceedings of the Seventeenth International Conference on Machine Learning, p.727-734, June 29-July 02, 2000

[11] E. J. H. Chang. "Echo Algorithms: Depth Parallel Operations on General Graphs," IEEE Trans. Software Eng., July 1982

[12] J.A. Nelder and R.Mead, "A simplex method for function minimization," Computer Journal, vol. 7, pp. 308–313, 1965.

[13] K.-F. Wong, S.-H. Chan, W.-C. Wong, Q. Zhang, W.-W. Zhu, and Y.-Q. Zhang, "Lateral Error Recovery for Application-Level Multicast," in Proceedings of IEEE Infocom'04, Mar. 2004, pp. 2708–18.

[14] http://www.isi.edu/nsnam/ns/

[15] http://www.cs.bu.edu/brite/

[16] Y. Chu, S. G. Rao, and H. Zhang, "A case for end system multicast," ACM SIGMETRICS' 00, June 2000.