LRing: A Layered Ring Topology for Reliable Streaming

Wen Jigang, Li Renfa, Xie Kun, Li Rui

LRing: A Layered Ring Topology for Reliable Streaming*

Wen Jigang, Li Renfa, Xie Kun, Li Rui
School of Computer and Communication, Hunan University
csjwen@comp.polyu.edu.hk

Abstract. To efficiently and stably deliver streaming media, researchers have developed technical solutions that are either based on a tree model or based on a mesh model. The disadvantage of tree-based model is limited stability and high cost, while the disadvantage of mesh-based model is low efficiency on streaming rate when nodes churn. In this paper, we propose LRing – A Layered Ring Topology for reliable streaming system, which well balance the maximum streaming rate and reliable streaming data in the system as well as achieve high utilization of node's cache. The simulation experiment demonstrates the effectiveness of LRing.

Keywords: media streaming, Layered Ring, P2P.

I. Introduction

Network stability and reliability are key issues for providing multiple services such as media streaming, network online games, subscriber services, etc. For video streaming on the wired Internet or wireless ad-hoc networks, traditional IP multicast has limited ability to meet QoS requirement of applications. P2P based streaming dissemination is an alternative and attracts many researches' eyes. Video-over-IP applications, e.g. Youtube [1], which are implemented based on P2P technology, have recently attracted a large number of users on the Internet.

In order to improve media streaming system performance and alleviate streaming source load, many P2P based overlay topologies for media dissemination have been deployed to provide on-demand and real-time video streaming over the Internet[2][3] and ad-hoc networks[4][5]. According to the overlay structure, P2P steaming systems can be classified into two categories: tree-based and mesh-based. Both of the overlay structures optimize several of aspects from different conditions and viewpoints. The tree-based systems, such as ESM [2], build a tree topology and push data actively from source to all of middle peers and leaf children peers. These systems mainly address the problem of how to choose parent peer in real-time dissemination scenarios to satisfy performance metrics such as optimal bandwidth utilization, minimum forwarding delay and minimum routing delay. However, tree-based system will lose its stability when peers dynamically join or leave, especially dynamic churn of upper peers in the tree. In a mesh-base streaming system, such as coolstreaming [3], pplive [6], peers are not confined to a static topology. Instead, the peering relationships are maintained based on

* This work was supported by the Ministry of Education. P.R.C for Streaming Research through the important project (108168).

the content availability and bandwidth availability on peers. A peer dynamically connects to a subnet of random peers in the system. The mesh-based systems are tolerant to peer churns, but fail to provide continuous quality of media streaming because of no guarantee for peer to effective dissemination. For some special environment, such as military, Business Corporation, and university, they need stable stream dissemination which are used for online war-gaming, important business video meeting and E-learning, respectively, but tree-based system and mesh-based system can't completely satisfy stable requirement of these applications.

Providing maximum streaming rate with stability, in other words, with strong redundant is difficult problem to deal with if using currently tree-based dissemination system or mesh-based dissemination. Because ring topology [11] has inherent characteristics of redundant, and its structure is easy to build, we choose ring as our basic infrastructure. Figure 1 shows an example of how streaming portions are transferred on the ring. There are six peers in the ring topology, represented by solid lines. Peer 1 is source peer, and peer 1, 3, 5 are power peers, which connect with each other to form a virtual ring, represented by dashed lines. Assume all nodes in the ring have infinite downloading capability but limited uploading capability. This assumption is close to real network conditions because some users may use ADSL/DSL connecting to the Internet and others use Ethernet, FTTH, etc. In Figure 1, we use bold lines and slim lines to represent uploading capacity difference between power nodes and normal nodes, and assume bold lines two times as great as slim lines in the bandwidth. As shown in figure 1, if uploading capacity of node 2 can't meet maximum streaming rate requirement of node 3, we need schedule and download from node 1 to reach maximum streaming rate. So, in the ring topology, how to organize peers to maximize the streaming rate of the whole ring system? This is our object in the paper to provide better performance. Moreover, some peers may disconnect from ring without any notice, or there is out of range of peer in ad-hoc networks because peer moves to other place. We use cache in the ring topology to provide reliable streaming service. In the figure, all peers cache some portions of the streaming file, trying to minimize system performance degradation. A naive solution is setting client's cache large enough to cache the entire streaming file, but this method does not comply with the fact: no one want to use most of its storage for caching a file especially, more probably, a large video file. In this paper, we will build a redundant model to cache file with the constraints of limit cache size of peers, which can provide reliable streaming.

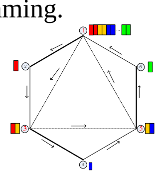

Fig.1. Example of data dissemination in the ring topology

To address the problems we mentioned above, we proposed a media streaming dissemination scheme called LRing, a layered ring topology. For the purpose of providing a stable media streaming service and guaranteeing the quality of service, we proposed a redundant model to build the layered ring topology. The topology can be used in wired and wireless ad-hoc networks.

In our ring based video streaming system, LRing adopts two levels: top ring level and base ring level. Every node in the base ring only has two connections with each other, in order to reach maximum streaming rate of the whole system, we also developed an optimal scheduling algorithm based on the redundant model. In the optimal algorithm, some nodes connect to other nodes on the ring with spare computing power, which can greatly decrease video stream playback delay. Using ring topology, LRing can effectively allocate node's uploading capacity to its descendant nodes. At the base ring level, all nodes are organized into computing-balanced group rings. Each group ring consists of a ring head and small number of nodes including power nodes and normal nodes. All of the ring head forming a top ring retrieves video from the video source server in P2P fashion; server only distributes video to group ring head; a normal node only maintains connections to its descendants in the same group ring; a power node maintains both connections with power node and normal node. LRing also has the ability to support a large number of peers with mild requirement on the number of connections on the server.

The paper is organized as follows. In Section II, we first formulate the problem and deduce the peer group ring strategy. Then we describe in details the system architecture and the peer management scheme. In Section III simulation results are presented. The paper is concluded in Section IV.

II. LRing: Layered Ring Topology for Reliable Streaming

As described in Section I, LRing is dedicated to form a stable video streaming dissemination system and offer good QoS to user even under dynamic churning. If allowing a large numbers of nodes to construct a single ring topology, it will introduce unnecessary maintenance overhead which making against for streaming data dissemination from the source to other nodes. Consequently, we need to limit the number of nodes in a single ring. Assume average delay between adjacent nodes is t_d, and the delay a user may accept when playback is t_u, and the maximum number of node n in a single ring should be t_u / t_d. For example, if t_d =50ms, t_u =1s, then n should not exceed 20 for performance consideration. Note that, in LRing, because we select power nodes to form additional rings, the delay from source to all of nodes on the ring may less than $n \times t_d$. The additional ring mentioned above is represented by virtual ring in this paper. Figure 2 illustrates a simple example of two-level LRing.

In figure 2, we only demonstrate a small unit of LRing. We call the small unit as mesh unit. LRing consists of many mesh units. A mesh unit includes two levels: base ring level and top ring level. In the base ring level, all nodes compose each group ring with limited number of node mentioned above. Each group ring includes a virtual ring consisting of selected power nodes which can speed up video delivery rate. At the top ring level, all group ring heads form one group ring. Video server distributes the content to all group ring heads then group ring head forwarding content to the rest of nodes in the group ring. When the number of nodes in a mesh unit is close to its maximum number which a mesh unit can support, it will split into two mesh units. For example, top ring is consisted of node *a*, *b*, *c*, *d*, *e*, and *f*, and the other nodes compose of base group rings. Streaming server node *a* distributes contents to the rest nodes of top group ring, then disseminate contents to base group ring nodes through the group ring heads. All of the group ring in figure 2 should build virtual ring to maximize streaming rate and cache some contents to provide reliable streaming.

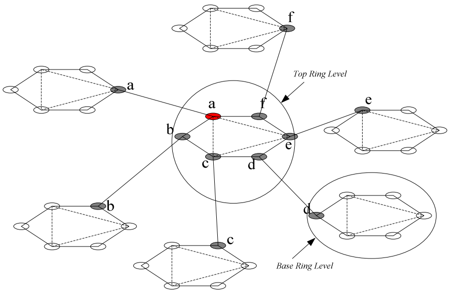

Fig.2. Architecture of a mesh unit

In the following, we firstly describe how to make strategies in LRing for reliable streaming. Then we illustrate how to manage the node to maintain LRing based on the strategy.

A. Node Grouping Strategy

At first, we will define some notions of each entity of LRing. Then we will discuss maximum streaming rate problem and a redundant model for node grouping strategy.

1) Formulate the problem

Nodes, the basic elements which form basic ring and redundant ring. Define $P = (P_1,P_2,\ldots,P_n)$. For each node $P_i(1\leqslant i\leqslant n)$ in system, its computing power can be quantized by three metrics: B_i denotes maximum bandwidth capacity; C_i denotes cache capacity; U_i denotes node delay.

Group ring, the basic ring which consists of nodes with closed physical location. Define Group=$(G_1,G_2,\ldots G_n)$. In LRing, all group rings is in the basic ring level.

Redundant ring, the ring connected by power nodes which selected from $G_i(1\leqslant i\leqslant n)$. Note that not all of the power nodes in the single ring are selected to form a redundant ring because we need to balance the group ring's distribution rate.

Mesh unit, the basic unit consisting of LRing. A mesh unit includes many group rings and redundant rings.

As mentioned above, LRing is a two level system which contains base ring level (*BR*) and top ring level (*TR*). Every node in BR has dual roles of providing streaming service to its descendant nodes as well as retrieving video data from its parent nodes. Because a ring topology may be benefit from the fact that bandwidth of a node may entirely occupied by its descendant node, there is a key problem of how to arrange peers to reach average maximum streaming rate in a heterogeneous environment where some nodes are powerful and others are normal. For example, in figure 1, node 2 is a normal power peer which capacity can't reach average value a dissemination system wanted. Consequently, we don't want node 2 to provide full streaming rate for node 3 to reach the maximum streaming rate, and that is why we should arrange nodes so that LRing can reach maximum streaming rate.

Assume there is a source node S_0, G group rings, and N nodes which cache capacity is C. Each group ring has V_g nodes, $g\in[1, G]$. Denote by $Min(r)$ the minimum streaming rate a user can accept. Denote by U_i the node i's upload capacity. Denote by U_{S0} the source node's upload capacity. Assume there is a redundant ring built on the group ring for reducing forward delay and speed up distribute rate at which user can playback. Therefore, denote S_i the number of nodes which serve content to node i. Denote R^{mesh} the maximum rate a mesh unit can approach. We formulate R^{mesh} as following optimization problem.

$$R^{mesh} = Max(U_{S0} + \sum_{i=1}^{N}\sum_{\{S_i\}} U_i)/N \tag{1}$$

subject to:

$$\sum_{\{S_i\}} U_i \geq Min(r) \tag{2}$$

$$S_i < V_g \tag{3}$$

The above formulation describes the optimal maximum streaming rate for a give LRing topology. Recall one main objective of this paper is to provide stable streaming service even when nodes dynamically churn. In LRing, we further give a redundant model trying to minimize system performance degradation. Assuming a media file is divided into m chunks. Denote by p $(p<1)$ the failure probability. For arbitrary chunk j, there are r_j nodes in group ring cached it and the failure probability is $p^r{}_j$. Our objective is to find the distribution of media file, which can minimize the loss of system QoS when node dynamically churns. Thus, we have:

$$\{rj\} = avg \underset{\{n_j\}}{Min} \sum_{j=1}^{m} p^{n_j} \tag{4}$$

subject to:

$$\sum_{j=1}^{m} r_j \leq CV_g \tag{5}$$

The maximum streaming rate LRing can approach is the average sum of how much the source node uploading capacity as well as other nodes contributes to the system. This explains the Equation (1). In Equation (2), the maximum streaming rate can not lower than user's minimum acceptance rate. All nodes' cache capacity in a group ring must larger than the sum of chunks' cache. This explains the Equation (5).

2) Node grouping heuristics strategy

In reality, it is difficult to get the values of parameters such as S_i, p. we will present a heuristic distributed strategy to achieve such a distribution without knowing the values of these parameters. We do the same assumption appeared in [7], with cache capacity and node failure probability added. Assume the failure probability of node p complies with uniform distribution. For power node selection strategy, we choose a general cost function mentioned in [8], with mildly modified. As described in the following:

$$Pow_i = \frac{1}{r} \times f_i + \frac{r-1}{r} \times \sum_i \frac{f_i}{N} \quad (6)$$

The left part in Equation (6) represents node's power we can get, and the right part of the Equation represents the minimum received power value to which node contributed and average superfluous power respectively. User can refer to [8] for details of the function. In this paper, we care about how to arrange these power nodes.

For space saving, we only describe the key heuristics strategy based on the above assumption and discussion, please refer to [12] for more details:

- The power node of group ring should comply with uniform distribution. The virtual redundant ring should balance system throughput to reach maximum streaming rate.
- The group head's upload capacity should be as large as possible, and the group head should have the ability to cache the whole of the streaming file for scalability.
- Each node should carefully maintenance cache replacement algorithm individually. Different segments of the same object should be cached in different peers. Thus, the failure of individual peers can only affect part of the media object.

B. Dynamic Node Management

LRing has a Rendezvous Point (RP) to handle LRing topology and record power node, group ring head, etc. RP maintenance a power node list table which contains three parts: node IP address, node survival history and node capacity. IP address entry provides IP information so that newly joining node can exchange contents with power node. RP will check each entry periodically to ensure whether power node is still alive or not. If not, it means the node may fail or depart from network without any notice, and this entry should be deleted.

1) *Group ring location*: A node who wants to join LRing to retrieve video contents should locate group ring firstly. There are three factors, which are decision metrics, load balance among rings and maximum number constrains of a basic ring, needed to be considered before group ring locating.

Intuitively, we can use the following algorithm to address group ring locating problems. Based on web publishing service of RP, a new node can get latest list table of power node and group ring head from RP. Suppose node *i* and *list[j]* represent a new node and a power node list entry. Assume $L_{i\text{-}j}$ represents the delay difference between node *i* and *j*. For simplicity, node *i* want to choose node *j* with the smallest $L_{i\text{-}j}$ as its adjacent node if the ring doesn't exceed maximum number constraints. However, this simple strategy may result in load balance problem to ring head if there are amount of nodes that send connection requests at the same time. Denote $\triangle t$, which represents threshold. If $L_{i\text{-}j} < \triangle t$, then randomly choose node *j* from a set of nodes which satisfy the inequation. The idea behind this is to balance the ring head's performance. After querying for the ring head from adjacent node *j*, node *i* can send a join request to the ring head.

2) *Node join*: To handle node joining into a group ring, the ring head should take two things into consideration. One is avoiding maximum streaming rate degradation because ring balance may be broken. The other is for scalability if there are too many nodes in a mesh unit to accept node joining.

A new node *i* is always inserted near power node *j* it selected between two connected nodes. If these nodes are unbalanced, the group ring nodes suffer from increased latency. To avoid this situation, the ring head must balance the ring nodes. Each group ring node nearby node *j* will determine the capacity of its neighbors. If the capacities differ from a threshold *h*, the node will swap its position with its neighbor node, which is closer to the more distant inner ring node. When there is a new node need to join, the above procedure executed by exchanging special query and response system messages.

If a mesh unit reaches its maximum supported number of nodes when a node joins in the group ring, it splits into two mesh units. RP notices this situation by checking the record list it maintained periodically. Then RP send split information to nodes at the top ring level. Half number of nodes at the top ring level will depart from the original ring with attached node at the base ring level. A new node joins randomly a mesh unit after mesh unit are successfully separated.

3) *Node departure*: While a node wants to leave, it sends departure information to ring head and its neighbor nodes to rebuild connection. So the handling of peer departure is straight forward. The group ring head takes the peer off its cluster member list, and informs other peers in the same cluster about its departure. In case the departing node is the cluster head, the RP node selects the backup power node from existing nodes in the cluster as the new group ring head.

4) *Node cache maintenance*: Remember that LRing not only need to reach average maximum streaming rate, but also provide redundant streaming cache for QoS requirement. To achieve these goals, we modify a heuristic replacement policy of [9] as follows. Each node replaces both those media segments with diminishing popularities as they rarely get accessed and those popular media segments with too many copies being cached. As a result, nodes accessing media objects completely will cache the latter segments and evict the beginning segments of the objects because they are more popular and have more replicas in the system than the latter segments. Nodes that access only the beginning segments will cache the beginning segments. Thus, naturally, a node will cache only a few segments of each object it has accessed, while the segments of each object are distributed across many peers in the system according to their popularities, reducing the negative effects caused by peer failures. The above operation needs no extra data transferring except the data requested by users.

III. Simulation Results

In this section we use ns2 to simulate LRing, and demonstrate that LRing is able to provide highly stability with acceptable maximum streaming rate. Firstly, we use GT-ITM topology generator tool to build a Transit-Stub topology with 250 nodes. We randomly choose one node as source node and 10 nodes to form a group ring. The arriving time and leaving time of a node complies with uniform distribution. We set nodes' upload capacity to follow up a distribution mentioned in [10]: 100 kbps for 20% of nodes; 200kbps for 40% of nodes; 500kbps for 25% of nodes; 2Mbps for 15% of nodes. In addition, the cache size of each node is set to 50M, which is enough for caching a video file if a group ring consists of 15-20 nodes. To avoid small probability event, every simulation runs 5 times, and we get the average results.

A. Performance evaluation criteria

In general, we use the following metrics to evaluate LRing's performance: *Path delay* between a pair of nodes is defined as the ratio of the end-to-end delay (measured in terms of either latency or number of physical hops) along a path connecting the two members on the overlay network; *Average streaming rate* is defined as total streaming rate of the LRing normalized by the number of nodes. *Average request failure ratio* is defined as the failed number of request normalized by the total number of request for a file chunk.

B. Experimental result

Figure 3 illustrates average transfer delay. For a single ring without any virtual

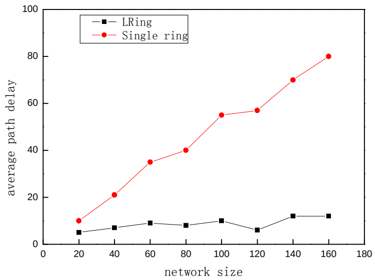

Fig.3. Average delay Vs network size.

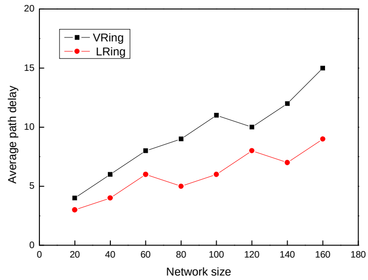

Fig.4. Average path delay Vs network size.

rings between power nodes, there is larger delay than LRing. As shown in Figure 3, using LRing leads to up to 85% reduction in the average transfer delay. This is because LRing has virtual ring creating short paths to the receiver on the network. Figure 4 illustrates average path delay between VRing [11] and LRing under different network sizes. With the network size increased, the average path delay of VRing rise faster than LRing. This is because LRing balance the structure of each group ring with the help of power nodes while nodes join or depart. However, VRing just build virtual ring without balancing load between nodes.

Figure 5 shows the average streaming rate of LRing with the time ranging from 0-24 hour. We firstly compute an optimal maximum streaming rate curve based on a real trace file, which is collected from PPLive, May 3, 2006. The maximum streaming rate of LRing is simulated by ns2. The LRing streaming rate is within 90% of the optimal upper bound all the time. The results in Fig. 6 show that LRing can significantly improve the QoS of media streaming and the average request failure rate with the increase of node cache size. For example, compared with the VRing, our system can reduce up to 82 percent failure requests. This is because more media data can be cached in the system due to the more storage contributed by nodes and, thus, improves QoS of media streaming.

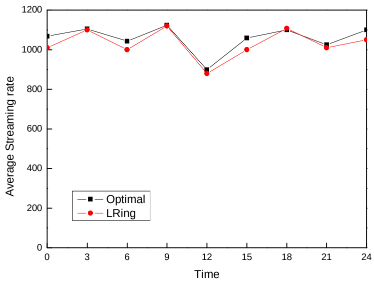

Fig. 5. Average streaming rate Vs Time.

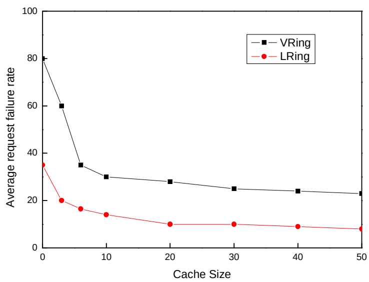

Fig. 6. Average request failure rate Vs Cache size.

IV. Conclusions

Existing Internet streaming media delivering techniques are either based on a tree model, or based on a mesh model. The disadvantage of tree model is its limited stability and high cost, while the disadvantage of a mesh model is its low efficient on streaming rate when nodes churn. In this study, we propose a layered ring system to address these limitations. In our system, nodes are organized into ring topology which structure is simple to maintain. We limit the size of a group ring for scalability consideration. In addition, we introduced a redundant model for the collaboration between clients, making the entire streaming media system performance-effective. The experiment shows that LRing can achieve 18% improvement in average request failure ratio compared to VRing systems.

References

[1] Youtube, "Youtube Homepage", http://www.youtube.com

[2] Chu, Y., Rao, S. G., and Zhang, H. 2000. "A case for end system multicast", *In Proceedings of the ACM SIGMETRICS* , 2000

[3] Xinyan Zhang, Jiangchuan Liu, Bo Li, Yum, Y.-S.P. "CoolStreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming", *In Proceedings of IEEE INFOCOM, 2005*

[4] Karumanchi, G, Muralidharan, S, Prakash, R. "Information dissemination in partitionable mobile ad hoc networks ", *In Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems*, 1999

[5] Baochun Li, Wang, K.H. "NonStop: continuous multimedia streaming in wireless ad hoc networks with node mobility," *IEEE Journal on Selected Areas in Communications,* vol.21, no.10, pp. 1627-1641

[6] PPLive, "PPLive Homepage", http://www.pplive.com

[7] Chao Liang, Yang Guo, Yong Liu. "Hierarchically Clustered P2P Streaming System", *In Proceedings of the IEEE GLOBECOM,* 2007

[8] Chu, Y, Chuang, J, and Zhang, H. "A case for taxation in peer-to-peer streaming broadcast", *In Proceedings of the ACM SIGCOMM*, 2004
[9] Guo, L, Chen, S, Zhang, X. "Design and evaluation of a scalable and reliable P2P assisted proxy for on-demand streaming media delivery", *IEEE Transactions on Knowledge and Data Engineering*, vol.18, no.5, pp. 669-682
[10] B. Fan, D.-M. Chiu, and J. Liu. "The Delicate Tradeoffs in BitTorrent-like File Sharing Protocol Design", *In Proceedings of ICNP*, 2006.
[11] Ahmed, S, William, Y, and Jennifer, C.H. "VRing: A Case for Building Application-Layer Multicast Rings (Rather Than Trees) ", *IEEE Computer Society*, 2004
[12] J.Wen, K. Xie, R. Li, "Adaptive Power Node Management Scheme in Ring-based Scalable Media Streaming System", *the HK polyu U., Tech. Rep.*, 2008