K-overlay: A Kautz Tree Structure for Video on Demand System.

Ubaid Abbasi and Toufik Ahmed
CNRS LaBRI Lab. – University of Bordeaux, France
{abbasi, tad}@labri.fr

Abstract— **In order to improve scalability and reduce maintenance overhead for Peer-to-Peer system, several architecture with constant degree and optimal diameter are proposed. However, the expected topology doesn't effectively utilize the bandwidth capacity of peers. In this work, we propose K-overlay, an overlay scheme based on unbalanced Kautz graph with $logd_n$ *diameter* and constant in-degree. We define the degree of a digraph as the maximum number of arcs arriving at or leaving from any vertex. The diameter of a graph is the number of arc traversals that is sufficient to reach any vertex from any other vertex. K-overlay structure is based on two-fold mechanism. (1) Organization of peers in concentric circles to maximize the outgoing bandwidth of peers (2) Content delivery through parent as well as neighboring peers. Through formal analysis and comprehensive simulations, we show that our proposed architecture achieves optimal diameter and good connectivity as compared to existing overlay architecture like P2Cast.**

Keywords: ***P2P Network, Kautz graph, Overlay Organization, Quality of Service (QoS).***

I. INTRODUCTION

To meet the demand of explosively growing multimedia applications, media streaming has been a research topic attracting significant interests over the past two decades. The ultimate goal of Internet media streaming is to satisfy the application requirements of as many end users as possible, with sustainable server bandwidth costs. The traditional client/server architecture advocates the use of large data centres to maintain streaming to end users at a large scale. The bandwidth costs on servers increases rapidly as the user population increases, and may not be manageable in corporations with limited resources. For instance, the bandwidth provisioning cost of YouTube servers are estimated at more than $1M per day [1].

IP multicast [2] and content delivery networks (CDNs), attempted to tackle the problem by conserving resources in the edge or core routers, or by load balancing across a large number of edge servers. However, the problem of scalability to a large user population in media streaming systems is only mitigated to a certain degree, not solved.

Over the past few years, Peer-to-Peer (P2P) networks have emerged as a promising approach for distribution of multimedia contents over a large scale network. P2P networks propose a different architectural design perspective, it offloads part of the bandwidth burden from dedicated streaming servers hosted by the content providers, and shifts them to end hosts themselves when they serve content to each other. The total volume of bandwidth consumed is not reduced, and in most cases, is even increased due to overhead of protocol induced messaging and redundancy. However the bandwidth burden is distributed across the network among the peers, which can contribute, relay, and consume the data.

Existing peer-assisted media streaming systems can be divided into two categories, live and on-demand media streaming, with the latter often referred to as video-on-demand (VoD) in the literature. The peer assisted VoD system typically contains three elements as shown in Fig. 1.

I. A number of dedicated streaming servers that serve media content, and can be considered as a single dedicated media source.

II. One or a small number of index servers that keeps track of state information of the system, such as existing peers. It is often referred as a Tracker.

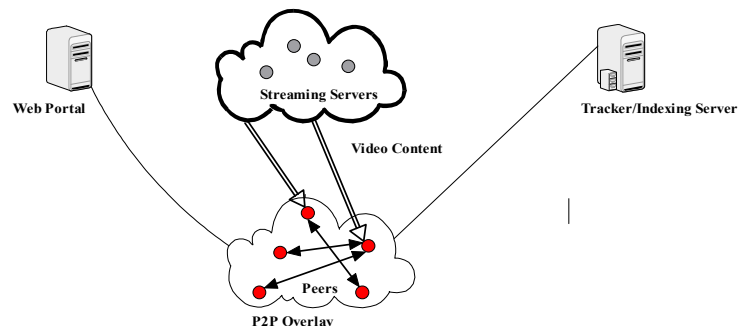III. A web portal that provides information about media channels/Video Content.

Figure 1: Architecture of P2P VoD System

Fundamentally, architectural and protocol designs in peer-assisted media streaming systems need to successfully address three characteristics observed in real-world streaming systems:

Scalability: A real-world peer-assisted streaming system needs to support at least hundreds of thousands of users and thousands of media channels simultaneously, with a reasonable playback quality to each of the end users.

UUSee Inc. experienced a flash crowd of 871,000 peers on the Chinese New Year Eve in 2007 [3].

Dynamics: The peer-assisted media streaming have shown a high level of dynamics. Peers who serve media content to one another may join or leave the network at any time. Network bandwidth between a pair of peers varies over time, sometimes quite drastically due to the shifting of bandwidth bottlenecks. Such dynamics challenge the design of any system architecture or protocol. Recent measurement studies have clearly shown that peer dynamics represent one of the most influential factors that affect the overall performance of peer assisted streaming systems [4].

Heterogeneity: According to a large set of live streaming traces obtained from the Cool Streaming system [5], it has been discovered that there exists a highly skewed distribution with respect to upload bandwidth contributions from peers. Such heterogeneity may have significant implications on resource allocation in peer-assisted streaming systems.

In this study, we aim at designing a novel P2P architecture with smaller diameter and peer degree even the number of peers is an arbitrary value. Thus, the scheme is practically easy to implement without being restricted by peer dynamics. We propose a robust and efficient P2P network, based on *unbalanced Kautz tree.* It achieves the optimal network diameter in a dynamic environment. The main contributions of this paper are as follows:

I.   We propose an un-balanced Kautz tree structure, an effective and robust P2P architecture which retains desirable properties of static Kautz digraph, such as optimal diameter.

II.   We design some algorithms to deal with peer join/departure, and topology changes.

III. We evaluate the topology properties of our architecture, delay and message cost of basic operations through formal analysis and comprehensive simulations.

The rest of the paper is organized in different sections. A brief related work and motivation is presented in section 2. Section 3 presents a brief overview of Kautz graph. The proposed architecture and its analysis are described in detail in section 4 and 5. Section 6 illustrates the performance evaluation and section 7 presents a brief conclusion while highlighting some of the future perspectives.

II.   RELATED WORKS

There have been tremendous efforts in the design and experimentation of video streaming systems in the past two decades, yet no single system has delivered the expected scalability and QoS. In this section, we highlight some of the known solutions that exploit the overlay organization for effective content distribution.

CoolStreaming/DONet (CS) [5] presents a framework for media streaming based on data-driven overlay networks where peers periodically exchange data availability information with each other and retrieves the unavailable data. This framework has shown great improvement in the video distribution with high scalability however it has two main drawbacks: (1) long initial startup delay due to random peer selection process and (2) higher failure rate in joining a program during flash crowd.

PRIME [6] presents receiver-centric video streaming mechanism over P2P mesh networks. The main objective of this system resides in minimizing the effects of bandwidth and content bottleneck. GridMedia [7] is well known system offering P2P-based IPTV services. It organizes the peers in unstructured overlay networks and implements a push-pull based approach to fetch the media contents from the neighbor peers. The iGridMedia [8] focuses on providing delay-guaranteed services to support real-time applications for large number of users. FissionE [9] utilizes kautz graph K (2, D) as its initial topology. The node identifiers are kautz strings with base = 2 and the lengths of node identifiers may be different. The initial kautz graph of FissionE, however, can only have an average degree of 4. FissionE does not support a higher degree, and thus cannot achieve a better performance even if there is a sufficient bandwidth for more maintenance overhead. Moore [10] was proposed with the goal of designing an efficient P2P scheme based on kautz graphs with arbitrary size and base. To this end, it proposed the incomplete Kautz digraph by applying the partial line graph (PLG) technique to kautz graphs. Furthermore, in order to select the ideal edge set for constructing incomplete Kautz digraphs with connectivity close to d, it adopt the Kautz factorization technique.

At the same time de Bruijn graphs have been proposed for peer-to-peer networks [11] [12]. Koorde [12] is an embedding of the de Bruijn graph into a Chord [13] network, it therefore inherits many properties from Chord. To gain resistance against node failures, either a number of backup links are introduced or the alphabet size K is increased up to O *(log n)*. Since the number of possible de Bruijn nodes is larger than the number of nodes present in the network, each Koorde node is responsible for a large number of de Bruijn identifiers. In [11] the de Bruijn topology is used to establish a distributed hash table which strives for probabilistic guarantees for lookups. Since the lookup guarantees are loosened, the routing table maintenance overhead is reduced. Furthermore, lookup hot spots can be alleviated by means of caching, and key collisions (which may occur e. g. in file sharing networks) can be tolerated at the cost of more lookup steps.

III.   Kautz Graph Overview

In graph theory a digraph is a directed graph, meaning that the arc between one vertex and another allows communication in only one direction. Bidirectional paths can be represented in a digraph by a pair of arcs, one running in each direction. Next the degree of a digraph is the maximum number of arcs arriving at or leaving from any vertex. The diameter of a graph is the number of arc traversals that is sufficient to reach any vertex from any other. In addition a small diameter is correlated with a small average number of hops, which directly reduces the contention for bandwidth among multiple concurrent message transfers.

Definition 1: A balanced kautz tree with depth k is a rooted tree. The root node has $d+1$ child nodes, while other nodes can have at most d children. Each node is assigned a unique identifier consisting of letters of a given alphabet and having a certain length L. The length of the identifier is also known as the level of the node or graph.

Definition 2: A d-ary Kautz tree with depth k is balanced if all leaf nodes are at the level k. A balanced kautz tree is a complete Kautz tree $KTree (d,k)$ only if the parent node of any leaf node has full child nodes, otherwise it is an incomplete kautz tree $IKT (d,k,n)$ with n leaf nodes.

Definition 3: For any node m, its predecessor is the last existing node anti-clockwise from it in a Kautz ring of existing nodes at same level, and its successor is the first existing node clockwise from in the same Kautz ring.

The out-degree and in-degree of a node is the number of out-neighbors and in-neighbors respectively. A graph G has constant out-degree (or constant in-degree) if for all nodes x in G the out-degree (or in-degree) of x is a constant. In Kautz-graph each node has a unique identifier consisting of letters of a given alphabet and having a certain length L. The size of the alphabet is called the "kautz-basis" and equals the out degree of each node. The length of the identifier is also known as the level of the node or graph.

Kautz digraphs have been discussed in the mathematics literature from many years. They have received a lot of attention because they are among the largest known graphs of a given diameter and degree. Every node in a degree d kautz digraph has d independent routes to every other node. This property of kautz graph is highly favourable for peer to peer networks where nodes join or leave the networks without a prior notification. The initial attraction of the kautz graph for many people is its logarithmic diameter. Unlike mesh or torus topologies, whose diameter is proportional to the square or cube root of the size of the network, the diameter of a kautz graph grows as the logarithm of its size.

IV.   K-Overlay Structure

The proposed overlay structure comprises of three entities: (a) Source peer, (b) Tracker, (c) Ordinary peers. The source peer is the seeder of the original video content. Tracker plays an important role in the proposed overlay structure that is responsible for assisting participating peers to communicate with each others to find their bootstrapping peers. When a new peer enters the network, it provides its unique identity and bandwidth range [IP, BW] to tracker. The detailed structure is presented in the following sub-sections.

A.   Overlay Organization

The overlay structure using kautz graph is shown in Fig. 2 with depth k that is equal to 3. Each peer has assigned a unique identifier. At depth 2, the last digit of label shows the parent identity. Similarly at depth 3 the last two digit of the label shows the parents identity. We used an unbalanced Kautz tree as an overlay structure. An unbalanced kautz tree has a fixed inbound but varying outbound. The reason of using unbalanced kautz tree structure is to utilize the bandwidth of different peers efficiently. Each peer maps to one leaf node in a balanced kautz tree. Each peer is connected to its immediate ancestor and predecessor. The peers in Kautz based overlay are organized into a circular structure according to their arrival. The peers are arranged from left to right according to their arrival.
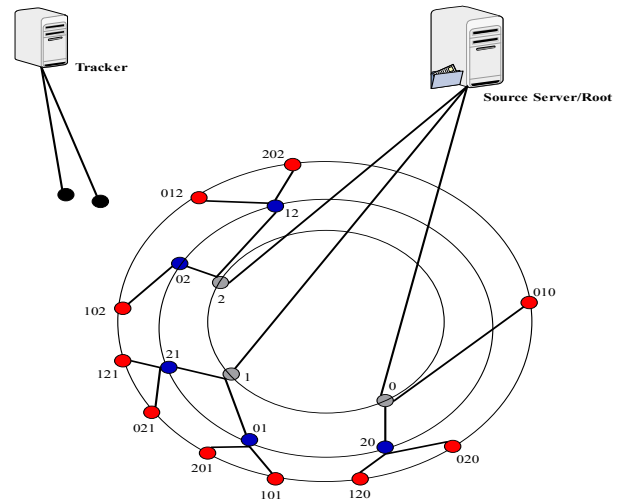

Figure 2: A rooted Kautz Tree with Depth 3

Each circular ring has varying number of peers in it. Hence peers are organized with minimum playhead distance with each other initially. The distribution of the neighbours in a circle has important influence on the performance of routing and member discovery. In VoD systems a peer needs many neighbours with close playhead so as to have plenty of choices for data

discovery. Furthermore, the peers connection with the previous and next circle assist in forwarding a "*seek query*" to a wide range more effectively. Each peer maintains $p+2$ neighbors apart from its connection with it parent, where p is the number of outgoing links of a peer i. The neighboring peers exchange the information of available segment in order to facilitate the exchange of data chunks.

Source Server is responsible for distributing video contents to a fixed number of ordinary peers that depends on the uplink bandwidth capacity of the source peer and the streaming rate of the video content. The degree *"D"* of the source server is given as:

$$\# \text{Degree } (D) \approx \frac{\text{Bandwidthout (source peer)}}{\text{Streaming rate}} \qquad \text{Eq.1}$$

Similarly the tracker calculates the degree of each parent using equation 1. In other word, the peer having greater bandwidth has higher number of child peers.

Tracker is also responsible for calculating the peer's contribution to the overlay. On the basis of this contribution the peers are promoted to upper circles closer to source. Tracker periodically performs the active measurement to calculate the serving strength (SS) of every peer. Serving strength is defined as the ratio of upload capacity to download capacity. This "SS" is further used to determine the contribution ratio (CR) of each peer (P_i) present in the K-overlay.

$$\text{Contribution Ratio } P_i \text{ (CR)} = \frac{\text{Serving strength}}{\text{Distance(source}, P_i)} \qquad \text{Eq.2}$$

The distance of a peer (P_i) from the source can be its delay time that is the time taken by packets to arrive at the peer after the source has streamed out. Eq. 2 shows that the peer with greater *CR* value will exhibit the best serving capacity and more localized to source peer. The contribution ratio based selection of peer also introduced the incentive mechanism. Peers providing greater contribution to the overlay network are selected and thus, they are able to receive contents directly from source peers. This mechanism encourages, ordinary peers to contribute more to the overlay network.

The *CR* estimation is carried out by every peer and obtained information is sent to tracker periodically. This *CR* estimation may not be feasible due to its excessive overhead when there is large number of peers present in the network because every peer performs such calculations. Thus, we consider the ordinary peers having *CR* value greater than a certain threshold to limit the overhead. Tracker regularly updates the source peers about new super peers present in the network.

B. Peer Joining & Departure

When a new peer requests the tracker to join the network, tracker in return provides the address of parent peer. The peer sends "*join*" request to the parent peer as provided by tracker. On receiving the response the new peer joins the network. If there is no response from parent peer within certain time threshold, peer sends a new parent request to tracker unless peer is successfully connected to parent. The new peer is assigned a unique identifier once it joins the network. For this purpose an extended version of KautzHash [9] is used to generate identifiers with varying base, which are uniformly distributed in the Kautz namespace. Tracker is also responsible for providing the address of the immediate predecessor and successor. The peer within a same ring exchange the buffer map records with its immediate predecessor and successor. Moreover buffer map is also exchange with parent and child peers. The tracker organizes the peers in clockwise order in the kautz ring. If all leaf nodes are allocated, the tracker starts the topology expanding process. Similarly it may start topology shrink process when required. Once, the new peer joins the network its parent peer updates the tracker that keeps track of number of peers joining any parent peer. This information is used for the future reference because each peer can support only a limited number of child peers, depending on outgoing bandwidth of the parent peer.

Ideally when a peer leaves the network, it initiates a departure message to its parent peer that releases the outgoing bandwidth and notifies the tracker for the graceful departure of this particular peer. The other case is not trivial, when a peer suddenly leaves the network without any prior notification. When a peer doesn't receive a buffer map information record within a certain time period from its parent or neighboring peer, it notifies the tracker. If there is no response received from a peer with in significant amount of time, it is assumed that the particular peer is no more available in the network. Tracker removes the address of the peer from the list and initiates the topology shrink process. During the shrink process tracker promote peers with greater CR value to upper circles.

V. FORMAL ANALYSIS OF K-OVERLAY

Lemma 1. In a D-Kautz graph G with base d and size N, the diameter of G satisfies $D < 2 \log_d N$.

Proof. Let $D(u, v)$ denote the distance between arbitrary two nodes $u = u_1u_2 \ldots u_m$ and $v = v_1v_2 \ldots v_n$ in graph G. Suppose $M(u, v) = i$ is the matching index. The matching index of two Kautz strings $u = u_1 u_2...u_m$ and $v = v_1 v_2...v_n$ represented by $M(u, v)$, is the maximum value of integer i, $0 \le i \le \min(m,n)$ which satisfies $u_{m-i+j} = v_j$ for any $j, 1 \le j \le i$. Let

$$w = u \oplus v = u_1 u_2 \ldots u_m v_{i+1} v_{i+2} \ldots v_n$$

In D-Kautz graphs the shortest path routing algorithm is adopted to route from u to v. The jth intermediate node s

in the path is of the form $s = u_k u_{k+1} . . . u_m v_{i+1} v_{i+2} . . . v_{i+j}$, a substring of w. Therefore, at the $(n - i)$ th step, we have $s = v$. Then we have $D(u, v) = |v| - M(u, v) \leq |v|$.

Let y be the node with the longest identifier in G. Clearly the diameter D of graph G satisfies $D \leq |y|$. On the other hand, the size of static Kautz graph $K(d,D)$ is $N = d^D + d^{D-1}$. Let x be the node with the shortest identifier in graph G.

Clearly for D-Kautz graphs, we have $N \succ d^x$. Thus the identifier length $|x|$ of node x satisfies $|x| \prec \log_d N$. Clearly the distance from y to x satisfies $D(y, x) = |x| - M(y, x) \prec |x|$. Therefore for each intermediate node $u = u_1 u_2 ... u_p$ in the path from y to x, the next hop v is of the form $v = u_t . . . u_2 u_3 . . . u_{pB}$ with $1 \leq t \leq 3$. Thus $|u| - |v| \leq 1$. Clearly the identifier lengths of nodes y and x satisfy $|y| - |x| \leq D(y, x)$. Thus the diameter of graph G satisfies $D < 2 \log_d N$. This completes the proof.

VI. PERFORMANCE EVALUATION

This section describes the performance evaluations of proposed overlay structure for different QoS parameters using NS-2 simulations [14].

A. Simulation Setup

We performed intensive simulations to study the performance of K-overlay in order to quantify the overall QoS parameters against P2Cast [15]. In P2Cast peers are organized in a tree structure, where each peer receive the requested stream and forwards the streams to other peers. In P2Cast, peers are distributed into sessions according to their playhead position.

Performance Metrics: The performance evaluation is carried for different QoS metrics that include: packet delivery ratio, jitter and average hop count. These parameters have significant role in determining the overall QoS for the streaming applications.

Network Topology: We used the BRITE universal topology generator [16] in the top-down hierarchical mode to map the physical network. The network topology consists of autonomous system (AS) and fixed number of routers. All AS are assumed to be in the Transit-Stub manner. Each topology consists of a number of autonomous systems and routers. The delay on inter-transit domains and intra-transit domains are assumed to be 90 ms and 40 ms respectively, while delay on stub-transit is assumed to be 30 ms and intra-stub transit links are randomly chosen between 5ms and 30ms. The incoming bandwidth of super and ordinary peers varies between 512 kbps to 5 Mbps and is uniformly distributed throughout the network. We deployed single source peer for providing original video content. Peers join the network at random intervals. We also introduced sudden peer departure at random

intervals. The presented results are the average results of the multiple runs of these simulations.

B. Results & Discussion

Fig. 3 shows the average hop count of the proposed mechanism in comparison with P2Cast. Hop count is referred as the number of intermediate peers involved on the overlay path from source to receiving peer. It is important to note that small overlay path doesnot necessarily guarantee small delay, however we want to emphasize the organization of overlay using K-overlay. It is observed that proposed mechanism have less number of hop counts due to organization of peers in Kautz tree structure.

Fig. 4 shows the average delivery ratio for K-overlay in comparison to P2Cast. Delivery ratio is defined as the number of packets delivered before deadline expires over total number of packets encoded. Unlike P2Cast, packets in K-overlay are received from parent peers as well as neighboring peers. A gap in sequence number identifies the missing packets. If a packet is obtained successfully within its playback deadline it is consumed otherwise dropped.

Moreover we evaluated jitter rate for both mechanisms in Fig. 5. The unavailability of packets at its playback time causes a jitter. K-overlay has comparatively lower jitter rate as compared to P2Cast. Sometimes an intermediate piece which cannot be fetched may increase the jitter rate. In K-overlay, chances of missing packets are relatively low due to organized content flow. If a packet is acquired by a circle, it implies that there are at least one or more sources for the contents. This flow of contents saves times as compared to P2Cast.

Fig. 6 shows the delivery latency experienced by data packets at different peers. We measure the average time for first packet arrival, for both mechanisms. We noticed that both models suffer from increased average time of first packet arrival, however, they stay relatively constant with increasing number of peers. Due to the existence of multiple neighbors, packets have higher chances to be delivered to the connected clients in a smaller period of time. The other reason is the systematic flow of content among different circles with near optimal overlay hops.

Fig. 7 shows the comparison of average delivery ratio with varying streaming rates. It is observed that as streaming rate increases, the delivery ratio for K-overlay remains relatively high. Even with the streaming rate greater than 500Kbps, delivery ratio is greater than 80%. This shows that network capacity of proposed mechanism is sufficient to support higher streaming rate.
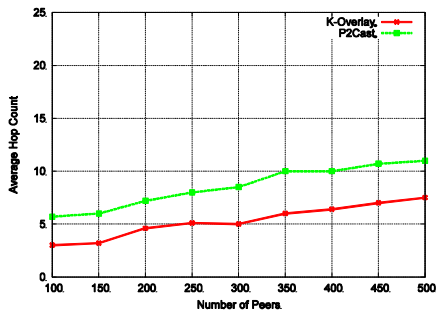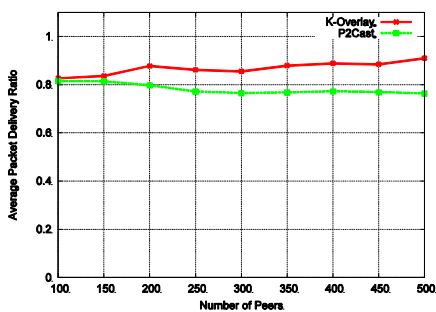
Figure 3: Average Hop Count
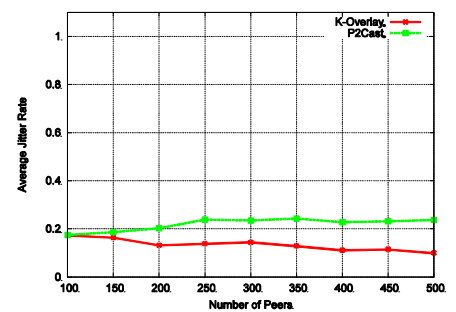

Figure 4: Average Packet Delivery Ratio
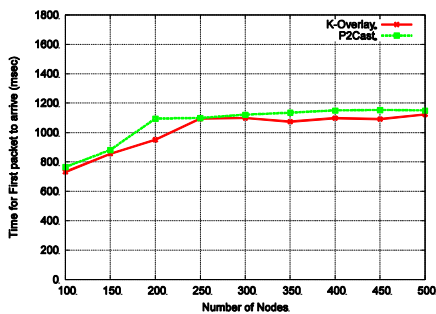

Figure 5: Average Jitter Rate
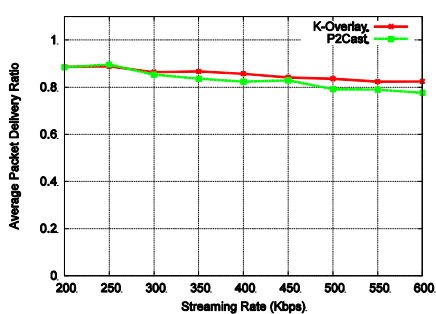

Figure 6: Average Packet Latency

Figure 7: Average PDR vs Streaming rate

V. CONCLUSION & FUTURE PERSPECTIVES

In this paper, we introduced a new overlay structure for VoD system namely K-overlay. It utilizes kautz tree structure for peer organization. The proposed structure focuses on effective utilization of outgoing bandwidth of peers in the network. Moreover it supports large number of peers with small diameter. K-overlay evaluation against P2Cast resulted in improved packet delivery and better video quality. Results also show a significant reduction of average number of overlay hops which leads to lowering the transmission delay. Finally, we believe that our results are promising and could provide research insight towards development of newer and efficient peering strategies in P2P media streaming systems.

REFERENCES

[1] Alleyne,Richard."YouTube:Overnight success has sparked a backlash". Daily Telegraph. http://www.telegraph.co.uk/news/uknews/2480280/YouTube-Overnight-success-has-sparked-a-backlash.html. Retrieved 2009-08-17.

[2] XING JIN, CHENG Kan-Leung, CHAN S.-H. Gary, "Island Multicast: Combining IP Multicast With Overlay Data Distribution", "IEEE transactions on multimedia , 2009, vol. 11, nº5, pp. 1024-1036

[3] Wu C, Li B, Zhao S (2007) Characterizing Peer-to-Peer Streaming Flows. IEEE Journal on Selected Areas in Communications 25(9):1612–1626, DOI 10.1109/JSAC.2007.071202.

[4] Li B, Xie S, Keung G, Liu J, Stoica I, Zhang H, Zhang X (2007) An Empirical Study of the CoolStreaming System. IEEE Journal on Selected Areas in Communications 25(9):1627–1639, DOI 10.1109/JSAC.2007.071203.

[5] Xinyan Zhang, Jiangchuan Liu, Bo Li, Tak-Shing Peter Yum, "CoolStreaming/DONet: A Data-driven Overlay Network for Efficient Live Media Streaming". In Proceedings of IEEE INFOCOM 2005.

[6] N. Magharei and R. Rejaie, "PRIME: Peer-to-Peer Receiver-driven Mesh-based Streaming".In Proc of IEEE Infocom 2007.

[7] M. Zhang, J-G. Luo, L. Zhao, and S. Yang, "A Peer-to-Peer Network for Live Media Streaming - Using a Push-Pull Approach". In Proc of ACM Multimedia 2005.

[8] M. Zhang, L. Sun, and S. Yang, "iGridMedia: Providing Delay-Guaranteed Peer-to-Peer Streaming Service on Internet". In Proc. of IEEE GLOBECOM 2008.

[9] Li D S, Lu X C. A novel constant degree and constant congestion DHT scheme for peer-to-peer networks. Sci China SerF-Inf Sci, 2005, 48(4): 421–436.

[10] Guo D K, Wu J, Chen H H, et al. Moore: an extendable peer to peer network based on incomplete Kautz digraph with constant degree. In: INFOCOM 2007. Anchorage: IEEE Press, 2007. 821–829.

[11] A.T.Gai and L.Viennot, "Broose: A practical distributed hash table based on the de-Bruijn topology", in *Peer-to-Peer Computing*. 2004, pp.167–164, IEEE Computer Society.

[12] M. F. Kaashoek and D. R. Karger, "Koorde: A simple degree optimal distributed hash table". In *Proceedings of the 2nd International Workshop on Peer-to-Peer System (IPTPS '03)*, Berkeley, CA, USA.

[13] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications". In *Proceedings of the SIGCOMM 2001 conference*. 2001, pp.149–160, ACM Press.

[14] The Network Simulator (NS-2), http://www.isi.edu/nsam/ns/.

[15] Y. Guo, K. Suh, J. Kurose, and D. Towsley, "P2Cast: Peer-to-peer Patching Scheme for VoD Service". In Proc of International conference on World Wide Web 2003.

[16] A. Medina, A. Lakhina, I. Matta, and J. Byers, 'BRITE: Universal Topology Generation from a Users Perspective, BU-CS-TR-2001-003, April 05, 2001.