MERIT: P2P Media Streaming with High Content Diversity and Low Delay

Shiyun Zhang, Abhishek Bhattacharya, Deng Pan, and Zhenyu Yang

School of Computing and Information Sciences
Florida International University, Miami, FL 33199
{szhan001,abhat002,pand,yangz}@cis.fiu.edu

Abstract. P2P is successful in various multimedia applications such as On-demand/live streaming due to the efficient upload bandwidth usage among participating peers which offloads server request thereby saving bandwidth as system size scales up. Many designs were proposed for P2P multimedia streaming systems, including the most promising tree/mesh overlays. In this paper, we propose *MERIT* as an integrated framework for scalable mesh-based P2P multi-streaming whose design objective is to preserve content diversity as well as optimizing start-up delay while satisfying the in-/out- bound bandwidth constraints. We formulate our design goals as an optimization problem and start with a centralized heuristic exploiting the global knowledge of peers. We then present a decentralized version of our algorithm which is scalable and follows similar design principles as the centralized one. Simulation results indicate that our heuristics outperform state-of-the-art approaches by improving streaming quality and start-up delay with efficient utilization of bandwidth resources at each peer.

Keywords: peer-to-peer systems, multi-streaming, delay management, content dissemination, bandwidth.

1 Introduction

Peer-to-peer (P2P) overlay networks provide a promising approach for live/on-demand streaming of multimedia content in comparison to traditional approaches due to the absence of IP multicast support and the limited scalability of client-server based model. Recently, P2P-based multi-streaming systems are gradually emerging with their potentials illustrated in the literature, e.g., [6,3,13]. Our work is formulated on a P2P multi-streaming i.e. MDC model consisting of a single source that transmits video content organized in independent multiple streams to a large set of receivers with the goal of delivering high quality media in a scalable fashion by effectively utilizing the contributed resources of participating peers. The content dissemination architecture is constructed using mesh-based overlay which is proved to have better performance due to its adaptability for network dynamics and bandwidth heterogeneity [7].

Some of the important objectives for designing an efficient high-bandwidth P2P multi-streaming system are listed as follows: (1) accommodate bandwidth

X. Zhang and D. Qiao (Eds.): QShine 2010, LNICST 74, pp. 530–543, 2011.

heterogeneity, e.g., peers with diverse up-/down- link bandwidth constraints, (2) maximize multi-streaming quality, e.g., the *content coverage* as the number of disjoint streams, (3) minimize start-up delay (STPD), i.e., the delay to each respective parent along the data delivery path. Our main contribution is to incorporate the importance of *content diversity* by preserving rare streams in MDC-layered streaming which helps to improve quality in the system as a whole. The concept of content diversification using rarest-first strategy is not new and already very popular in file-distribution applications invoked initially by BitTorrent [4]. We utilize this strategy to be an important component in MERIT framework which represents the first effort of providing a careful integration for all the above mentioned objectives to improve QoS of real-time P2P multi-streaming applications.

Figure 1 shows an instance of a mesh-based streaming session where the streaming server, S, is involved in disseminating the multi-stream content (divided into streams a, b, c, d) to the participating peers labeled as A, B, C, D, and E. For simplicity, we assume each stream takes similar bandwidth and the residual outgoing bandwidth is represented as the number of supportable streams. Meanwhile, all the overlay links are marked with delays. The figure also illustrates the streams received at each peer. Given that, the start-up delay (STPD) is calculated as the maximum delay from its respective parent over associated multiple delivery paths. For example, the start-up delays of peers in the figure are as follows: $\text{STPD}_A = 8$, $\text{STPD}_B = 11$, $\text{STPD}_C = 7$, $\text{STPD}_D = 5$, and $\text{STPD}_E = 13$.

As an illustration of the problem, consider the MDC-based multi-stream planning for peer E as we want to maximize its content coverage (number of disjoint streams received) by using rarest-first stream ordering while reducing the start-up delay. We assume that the incoming bandwidth for E is 2 (i.e. can support

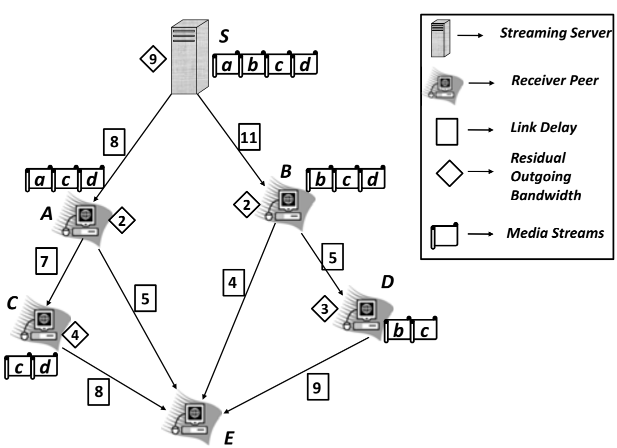

Fig. 1. A simple mesh overlay based P2P streaming model

a maximum of 2 streams). There are several choices depending on various optimization objectives. In [5], a greedy algorithm is proposed which always utilizes peers with maximal outgoing residual bandwidth. Thus, peer C will be selected as parent with the delivery path S→A→C→E achieving a content coverage of 2 (streams c, d from C) with a STPD of 8. In [9], a multiple-tree overlay based approach is proposed with the purpose of maintaining high streaming quality when nodes leave and join frequently. Since a parent is randomly chosen which has enough residual outgoing bandwidth to deliver a particular stream, the delivery paths could be S→A→C→E and S→B→D→E. The content coverage for this plan is 2 (c from D and d from C) with a startup delay of 9. We observe that currently there is no integrated solution which combines delay management, bandwidth constraints and rarity-ordered content coverage under one framework.

As motivated, we propose *MERIT* (media streaming with integrated framework) in this paper which aims at considering content coverage and start-up delay in an integrated fashion while satisfying the in-/out- bound bandwidth constraints. In the aforementioned example, the solution derived by *MERIT* will select A and B as parents with the delivery paths as S→A→E and S→B→E with a content coverage of 2 (a from A and b from B) and STPD of 5. *MERIT* consists of the delivery mesh construction and the content planning methods in a more tightly coupled manner to obtain better results than previously proposed schemes. Basically, we exploit rarest-first strategy and a *network-power* metric for decision making in the mesh construction and content planning stages to achieve optimal solutions. Simulation results further confirm the effectiveness of our solution in comparison with other state-of-the-art approaches.

The remaining paper is organized as follows. Related work is discussed in Section 2. The formal description of our problem is introduced in Section 3. Section 4 describes our proposed heuristic solution. Experimental evaluations are discussed in Section 5 and we conclude in Section 6.

2 Related Work

The concept of P2P streaming and its utility in video distribution was first proposed by [14]. Initial approaches involved the construction of a single multicasting tree overlay [12] and the content is distributed to all the nodes by organizing them into clusters/layers. Single tree approach faces some serious issues regarding vulnerability as the failure of an upper-level node cuts off the entire subtree and repairing the tree is difficult and costly with extreme conditions of dynamic peer participation. Multiple-tree approach was proposed in [2,9] to overcome this difficulty. [9] proposed a hybrid client/server architecture where the peers relay data to each other to protect the server thereby achieving scalability and robustness. [2] employs multiple description coding to break down the original stream into a number of sub-streams and then pushing each sub-stream through a specific stripe tree. The multiple trees are organized in an internal-node disjoint manner which achieves resiliency since the failure of a peer will only affect a single stripe tree thereby having lesser effect on streaming quality.

Mesh-based overlays were initially introduced as application layer multicast solutions but later it was incorporated in P2P video streaming domain and its effectiveness was proved over multiple tree approaches in [7]. The robustness of mesh overlays is mainly due to its randomized construction with each peer having multiple parent/children which helps for better adaptations during peer churn. According to related literature, research on mesh-based P2P streaming system can be divided into two important phases: (a) mesh construction [10,11], and (b) content scheduling [1,5]. [10] focused on the mesh construction problem and proposed a minimum delay mesh with the objective of minimizing end-to-end delay whereas [11] builds the mesh in a way for efficient utilization of peer bandwidth. On the other hand, [1] provided various selection schemes for push-based live streaming application and found a particular strategy to be optimal for both delay and rate whereas [5] proposed a content scheduling scheme for layered video streams for maximizing streaming quality. [15] is one of the commercially successful internet p2p live streaming systems which employs a mesh-based approach for connecting the peers and then selecting proper parents for receiving specific sub-streams. [7] also proposed a random mesh overlay by optimizing bandwidth-per-flow which will maximize the utilization of both incoming/outgoing bandwidth of all peers followed by a pull-based content dissemination mechanism.

Our main distinction from the previous work is that the organization of delivery mesh and content planning is done in a more integrated fashion with the desired objective of overarching rarity-ordered content coverage, start-up delay and bandwidth constraint under one common framework. We believe an integrated solution is critical for real-time multi-streaming applications.

3 Problem Formulation

We present a formal description of the problem which considers the delivery mesh construction and content planning phases across a set of receiving peers for streaming content from the server with asymmetric incoming and outgoing bandwidths. We introduce the following terminologies which will be helpful in formulating the problem:

- ***Mesh graph***, G=$\langle$V, E$\rangle$ is modeled as a directed overlay network where V is the set of vertices representing the server/peers and E is the set of overlay edges. We denote v_0 as the streaming server and $P = \{v_1, v_2, .., v_{n-1}\}$ the set of receiving peers. Each peer v_i possesses an incoming bandwidth I_i and an outgoing bandwidth O_i quantified as the number of supportable streams. For v_0, we only consider its outgoing bandwidth.
- ***Link Delay***, d_{ij} is associated with every edge $e_{i \to j} \in E$ and $v_i, v_j \in V$ which represents the underlying unicast path delay from peer v_i to peer v_j in the physical network.
- ***Candidate parents***, H_i is the set of all possible parents derived from the mesh overlay for each receiving peer v_i where $H_i \subseteq V$ and let $|H_i| = l_i$.

- ***Selected parents***, H_i' is defined as the set of chosen parents from H_i for receiving peer v_i such that $H_i' \subseteq H_i$ according to some content planning schemes while respecting the incoming/outgoing bandwidth constraints. Let $|H_i'| = m_i$.
- ***Content set***, $C = \{C_1, C_2, .., C_c\}$ is the set of total content streams generated at server v_0 and the number of total content streams is $|C| = c$. We assume the entire video is divided into substreams using Multiple Description Coding (MDC) such that each substream can be rendered independently with the overall video quality improves as the total number of disjoint substreams increases. However, with modification our scheme can also be used with layered coding such as SVC where the higher layers are dependent on the lower layers. As a general idea of integrating delay management, content coverage and bandwidth constraint, our proposed framework is not restricted to any specific coding technique.
- ***Content coverage***, CC_i is the set of content received by peer v_i and is defined as follows:

$$CC_i = \bigcup\nolimits_{\forall v_k \in H_i'} C_i^k \tag{1}$$

where C_i^k denotes the content set received by v_i from parent v_k and each set C_i^k is disjoint in CC_i (i.e., $C_i^a \bigcap C_i^b = \emptyset$ for $\forall v_a, v_b \in H_i'$). The ideal objective will be to achieve $CC_i = C$ for each v_i.
- ***Rarity Index***, RI_i for each peer v_i is defined as: Suppose each content stream C_i collected in CC_i has a *rarity factor* r_i^j ($j \leftarrow 1$ to $|CC_i|$) which is defined by the total number of parents in H_i' that currently possess stream C_i. The Rarity Index is defined as follows:

$$RI_i = \sum\nolimits_{\forall j \in CC_i} r_i^j \tag{2}$$

- ***Startup delay***, Startup delay D_i for each peer v_i is defined as the maximum link delay with its neighbor among the possible delivery paths and is formulated as follows:

$$D_i = max_{\forall v_j \in H_i'} d_{ji} \tag{3}$$

where v_j is a selected parent of v_i (define $D_0 = 0$).

Our problem is to construct a content delivery mesh (i.e., deriving H_i' for each peer v_i) and a content planning scheme (i.e., C_i^k) such that the following objectives are satisfied:

$$\textbf{minimize:} \quad \sum_{i=1}^{n-1} RI_i \tag{4}$$

$$\textbf{minimize:} \quad max_{\forall v_i \in P}(D_i) \tag{5}$$

but subjected to the following constraints:

$$\forall v_i \in P : |CC_i| \leq I_i \tag{6}$$

$$\forall v_k \in V : \sum_{v_i \in P} C_i^k \leq O_k \tag{7}$$

4 refers to the minimization of the rarity-ordered content coverage which basically ensures to preserve rare streams thus helping to promote content diversity, and 5 indicates the minimization of the average startup delay among all the peers. 6 and 7 refers to the inbound/outbound bandwidth constraints respectively.

4 Proposed Solution

Given a mesh graph G, our objective is to create a delivery mesh so that requirements of content, delay and bandwidth are fulfilled. We employ a pull-based or receiver driven content dissemination mechanism where the initial process consists of selecting m_i parents in H_i' from l_i candidate parents in H_i. The naive solution for the problem of choosing m_i from l_i is to try all possible combinations till we exhaust and then finding the optimal one, which is computationally intensive. To avoid such strained process, we devised a simple mechanism for our heuristic solution which is divided into two phases: (1) *Stream Selection Policy*, followed by (2) *Parent Selection Policy*. Next, we describe the details of the two phases with an illustrative example as shown in Figure 2.

The Stream Selection Policy enforces the rarest-first strategy for preserving content diversity. The policy is simple and described as follows: Compute the *rarity factor*, r_i^j of each stream by summing up the number of parents in H_i' currently possessing the particular stream. Now, sort them in ascending order so that the rarest stream is at the top of the list. The sorted stream list is passed to the Parent Selection Policy for picking suitable parents with desired streams. The Parent Selection Policy scans the sorted list from the top and schedules a

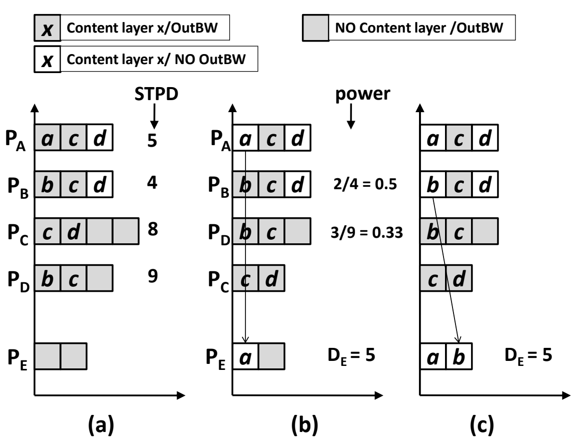

Fig. 2. Content planning in MERIT based on rarity and power ordering

single parent for each stream so that the content can be pulled. If there are more than one parent entry for a particular stream, then we need to make a decision for selecting one parent. In such situations, we utilize the concept of *network-power* as an optimization metric for ranking the parents in descending order and selecting the one with the highest power. Traditionally in networking, *power* is defined as the bandwidth divided by delay and is formulated as follows [10]:

$$pow_i^j = \frac{R_j \Uparrow}{d_{ji}} \tag{8}$$

where pow_i^j is the *power* of parent v_j with respect to child v_i, $R_j^{\Uparrow}$ is the residual outgoing bandwidth of parent v_j. The *power* metric essentially tends to prefer parents with higher available outgoing bandwidth and lower start-up delay.

Going back to the example in Figure 1, we show how this scheme translates the ordering to reach the solution computed by *MERIT* as illustrated in Figure 2. First, we compute the *rarity factor* of each stream with respect to peer E, namely, r_E^a, r_E^b, r_E^c, and r_E^d. For example, $r_E^a = 1$ since stream a is present with only parent A, and likewise, $r_E^b = 2$, $r_E^c = 4$, and $r_E^d = 3$. Next, we sort the stream list based on ascending order of *rarity factor* values, followed by the parent selection process. We can immediately assign stream a from parent A since there is only one provider. Since the in-coming bandwidth restriction for E is 2, we have chance to select one more stream for improving content coverage. *MERIT* will choose b from the sorted list since stream b is the second rarest content among the parents. Now, stream b is present with parents B and D, so we need to calculate the power (pow_E^B, pow_E^D) and select the parent with the highest value. For example, $pow_E^B = 0.5$ and $pow_E^D = 0.33$. Thus, B will be selected as the new parent and the final delivery paths computed are: S→A→E and S→B→E, as derived earlier in Section 1. The justification for this content planning mechanism is derived from the fact that it increases the content coverage by selecting the streams in the rarest-first order which will ensure content diversity in the whole system and thus, will lead to an overall improvement in streaming quality of all the peers. The *network-power* based parent selection mechanism tends to prefer parents with higher bandwidth and lower delay which will improve overall QoS by generating a high quality content delivery mesh.

4.1 Centralized Solution

In this section, we develop a centralized heuristic based on the complete knowledge of the peers and the entire mesh topology including bandwidth/delay for all peers/links. Our algorithm is shown in Table 1. The centralized algorithm is a greedy heuristic where it always chooses the parent hosting the stream with the highest *rarity factor* r_i^s as the first choice. The algorithm starts from the source and in each iteration it pushes the children (v_i) with the least startup delay that have not received any stream till now to expand the delivery mesh (Line 11). This is a favorable choice as it tries to greedily reduce the average startup delay among all the peers. In the content planning phase, peer v_i chooses

Table 1. Centralized MERIT algorithm

```
1.  Initialize a set, Suppliers ← ∅
2.  Suppliers←Suppliers∪{v_0}  // v_0 is source
3.  Initialize a priority queue, Neighbors ordered by
     increasing start-up delay
4.  Neighbors←Neighbors∪ children of v_0
5.  for each i←1 to n − 1
6.     R_i^⇑←O_i  // set residual outgoing BW to full
7.     R_i^⇓←I_i  // set residual incoming BW to full
8.  change←true
9.  while (change == true)
10.      change←false
11.      v_i←Neighbors.dequeue
12.      H_i← up-link neighbors of v_i∩Suppliers
13.      Intialize a candidate stream list Z_i
14.      for each v_j∈H_i
15.         for each stream s ∈ CC_j
16.            if(s ∉ Z_i)
17.               Z_i ← Z_i ∪ {s}
18.      Compute r_i^s for each s∈Z_i
19.      Sort Z_i in ascending order based on r_i^s values
20.      while(|CC_i| < R_i^⇓)
21.         Extract next s∈Z_i in sorted order
22.         if(r_i^s==1)//assume peer v_j have stream s
23.            parent←v_j
24.         else if(r_i^s > 1) //assume peer set X_i having
                               //stream s
25.            for each x∈X_i Compute pow_i^x
26.            Select parent←x with highest pow_i^x
27.         if(s∉CC_i) and R_parent^⇑≥1
28.            CC_i←CC_i∪{s}
29.            change ← true
30.            if (D_i > d_ji)  // update startup delay
31.               D_i←d_ji
32.            R_i^⇓ ← R_i^⇓ − 1
33.            R_j^⇑ ← R_j^⇑ − 1
34.      if (CC_i≠∅)
35.         Suppliers←Suppliers∪v_i
36.         Neighbors←Neighbors∪ children of v_i
```

streams with increasing rarity (Line 21). Initially, the rarest stream with least r_i^s value is chosen and the number of parent's currently hosting the stream is found. If a single parent only host the stream, then the parent is selected for stream s provided bandwidth constraints are met (Lines 22, 23). If there are multiple parent's hosting stream s, then pow_i^j is computed for every candidate parent j, and the parent with the highest *network-power* value is selected for stream s

(Lines 24 to 26). After that, the startup delays, the in-/out- bound bandwidths are updated (Lines 30 to 33). Peer v_i continues to select parents until no more streams can be allocated thereby maximizing content coverage/streaming quality. The algorithm keeps pushing peers and adding paths into the delivery mesh according to the order of *rarity* and *network-power* metrics. It terminates when all the peers have been tested for receiving and forwarding streams to all other neighboring peers.

4.2 Decentralized Solution

A scalable and practical solution cannot be centralized since we cannot afford to store and maintain the global information at the server which will be overburdened with queries from increasing number of peers in the system. Moreover, it also creates a single point of failure and bottleneck related problems as the size of the system increases. To accommodate such situations, we implement a decentralized solution which is realistic and scalable but follows the same guiding principles as the centralized protocol.

To start with, we assume the existence of a rendezvous point for facilitating the new peer to join the system. The rendezvous point keeps a small random subset of live peers in the system. A joining peer contacts the rendezvous point upon entry and is provided with a candidate list of live peers. The joining peer then probes all the candidate parents in the list and in return receives the information regarding content streams, residual outgoing bandwidth and average start-up delay from them. To avoid waiting too long for the unresponsive parents due to slow processing or network congestion/packet loss, the joining peer waits for a fixed interval of time and then it removes the parent from the candidate list. The joining peer calculates the *rarity factor* scores of each stream from the newly formed neighborhood of parent set with the information received. The joining peer sorts the streams based on decreasing *rarity factor* value and then follows a greedy selection method. The joining peer chooses the parent with the highest *network-power* score from Equation 8 and streams from it based on the bandwidth/content availability. If this parent cannot serve the joining peer, then the next parent is contacted and the process is repeated till either all the streams are received by the joining peer or there are no more new parents to scan. Once the parents are selected, the next part of content planning and delivery path generation is the same as the centralized protocol.

Peer leave can be of two types: (a) graceful leave, when the leaving peer informs its parents and children beforehand so that they can have sufficient time to readjust the delivery mesh as quickly as possible, (b) peer failure, when the peer voluntarily/involuntarily leaves the system without informing others. Failure is hard to overcome as it can happen at any time in any volume without prior information. To tackle this problem, each peer probes its parents and children at periodic intervals, and if any peer doesn't respond back within the specified period then it drops the respective entry from its list and frees the allocated resources. If a peer looses many parents and its streaming quality is sufficiently degraded, then it can query its neighbor's neighbor or the rendezvous point in search of potential parents.

We can observe that the initial peer selection for joining the mesh and stream selection with the highest rarity value in the whole system, are optimal choices by centralized protocol considering the global information availability. For a decentralized protocol, the selection space is pruned down to only the neighborhood parent set, which are sub-optimal choices in comparison to the centralized version. Still, a decentralized solution is desirable due to its higher system efficiency in requirement to maintain only a small set of neighborhood information. Moreover, the shortfall in QoS performance metrics of decentralized in comparison to centralized is within acceptable limits (as explored in the following section) making it a more suitable and practical choice.

5 Experimental Evaluation

In this section we discuss about the various simulation experiments that are performed to test the different algorithms under various scenarios. We use BRITE [8] to generate the topologies for our experiments. Each topology was generated from top-down hierarchical models with autonomous systems and routers. Peers are generated randomly and attached to the router nodes with the total number of peers varying from 100 to 1000. The access link bandwidths are set in the range of 100 kbps and 500 kbps with a mixture of exponential, uniform and heavy-tailed distributions. We intentionally limit the maximum bandwidth within 500 kbps and take a pessimistic approach to test the *MERIT* algorithms in resource scarce conditions. We keep the streaming rate of each stream at a constant of 100 kbps unless otherwise noted. The number of total streams is kept at a constant value of 6(unless otherwise specified) for all experiments. We plot all the results by taking an average of 10 runs for each point in the graph with a confidence interval of 95% and different BRITE topologies for each run. To evaluate the effectiveness of rarity in MDC-based layered streaming, we experimented with two different versions of decentralized MERIT: (1) MERIT_R i.e.,considering the rarity function as described in the pseudocode, and (2) MERIT_N i.e., replacing the rarity based stream selection with a randomized one to improve streaming quality and a power-based parent selection process. We compare our results with CoopNet (Multiple-tree approach as proposed in [9]). We build distribution trees as described in CoopNet and employ a random parent selection scheme respecting bandwidth constraints for each tree which allocates corresponding stream.

We first compare the content coverage (i.e. the total number of distinct streams received) of our approach with CoopNet as shown in Figure 3. Clearly, the content coverage of MERIT is better than CoopNet. The important observation in Figure 3 is that even MERIT_R performed better than MERIT_N which states the effectiveness of preserving rarity in MDC-based layered streaming solutions. Each point in the graph represents the average content coverage over the total number of peers. The average content coverage over all the different network sizes for MERIT_R comes to 4.499 (and 3.427 for MERIT_N) whereas

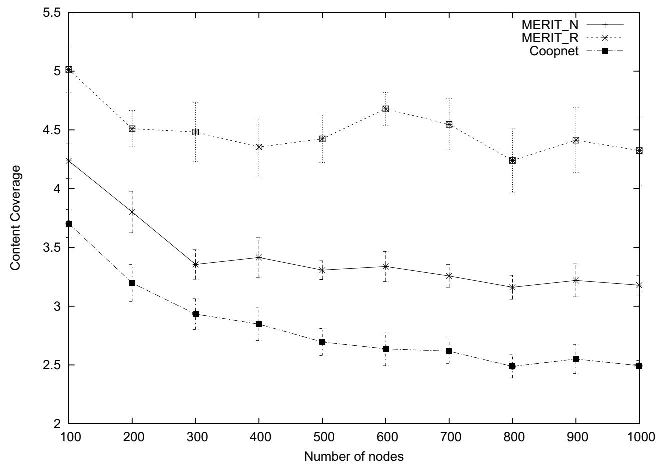

Fig. 3. Plot representing average content coverage over different network sizes

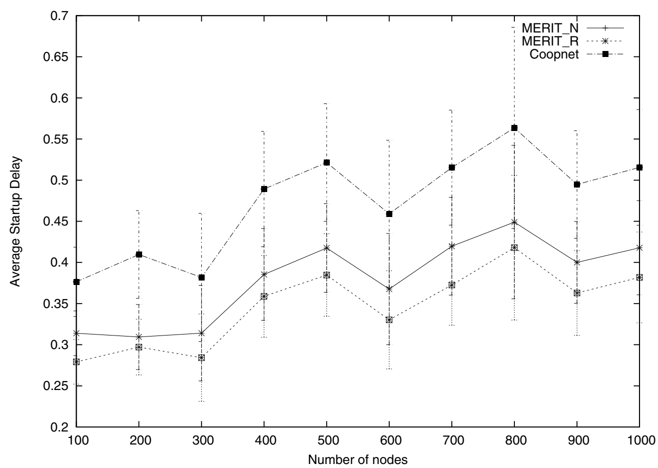

Fig. 4. Comparsion Plot of average start-up delay over different network sizes

CoopNet accounts for 2.816. The rarity-based MERIT_R achieves a 59.8% performance gain over CoopNet which proves that the concept of *content diveristy* and *network-power* helps to improve the streaming quality to a large extent. We also observe that the performance curve for MERIT_R does not drop and remain within acceptable limits with the increase in network size which indicates the scalability of the protocol.

Minimizing startup delay for each peer is one of MERIT's optimization objectives whereby the peers with low startup delay are given preference in joining the content delivery mesh and parent selection policy as evident in the expression of

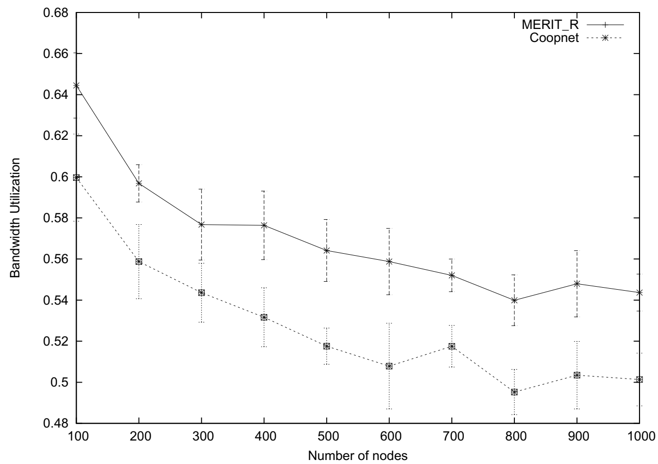

Fig. 5. Percentage bandwidth utilization over different network sizes

power. We plot the average startup delay achieved by each algorithm in Figure 4 where each point in the graph represents the value collected from all peers. From Figure 4, it is clear that the start-up delay of MERIT_R and MERIT_N is better than CoopNet which is quite natural since CoopNet does not make any attempt to reduce start-up delay explicitly. The *power* based parent selection helps to improve the start-up delay to a considerable extent which is an important objective for live, interactive streaming applications. MERIT_R achieves about 24.8% startup delay reduction over CoopNet. The average startup delay across all the network sizes for MERIT_R is 0.379 and for MERIT_N it comes to 0.346 which is comparatively better than that of CoopNet at 0.473. MERIT_R and MERIT_N are within comparable limits since rarity does not help in improving average start-up delay.

We plot the percentage of bandwidth utilization of MERIT_R and CoopNet over different network sizes as shown in Figure 5. It is observed that outgoing bandwidth is a scarce resource and needs to be efficiently utilized by the system. On the other hand, the incoming bandwidth is more abundant since a peer generally has its incoming bandwidth to be multiple times of its outgoing bandwidth. In our experimental setting we have assigned incoming/outgoing bandwidths to be between 100 kbps and 500 kbps and so it will be more relevant to compute the aggregated outgoing/incoming bandwidth utilization. The average bandwidth utilization factor for MERIT_R is around 58% which is generally acceptable and does not seem to vary much by increasing network size. The average bandwidth utilization for CoopNet comes to 52.7% indicating an average of 5.3% performance enhancement of MERIT over CoopNet. We are currently investigating further to improve the bandwidth utilization and more even distribution among the peers which will help to uniformly share the streaming quality.

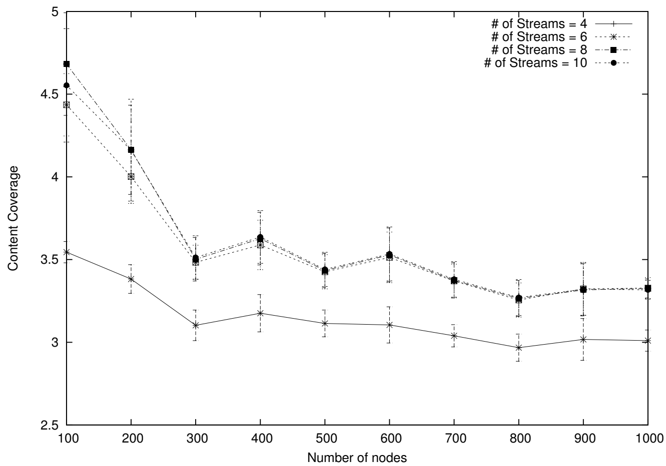

Fig. 6. Plot of streaming quality for number of layers=4, 6, 8, 10

Next, we investigate the effect of number of streams on content coverage for centralized MERIT with results plotted in Figure 6. Increasing the number of streams will provide diversification of content availability but places more demand on the system as each peer will strive to receive the maximum number of streams. Analyzing Figure 6, we observe that the percentage of content coverage decreases consistently with the increase in the number of streams. We discover that the percentage of content coverage with 4 streams is 78.6% which decreases to 59.5% for 6 streams and further decreases to 45.3% with 8 streams and 36.1% with 10 streams for constant bandwidth restrictions of 100-500 kbps. So, increasing the number of streams generates more system load in MERIT which reacts by pulling down the streaming quality of the peers.

6 Conclusion

In this paper, we presented MERIT, which is an integrated mesh-based P2P media dissemination solution under the scenario of MDC-based multi-streaming and heterogeneous resource distribution of peers. We argued the importance of content rarity preserving property in layered media streaming application which helps to improve diversity, thus achieving an overall improvement in quality over all peers as verified through simulation studies. Moreover, we also incorporated the concept of network-power to improve the start-up delay which is a crucial parameter for live interactive applications. We proposed the MERIT framework composed of Stream Selection Policy and Parent Selection Policy to generate a high-quality content delivery mesh in layered media streaming solutions. We have described a formal framework of the problem and the various desired objectives with constraints that are required to be met. Based on our approach, we proposed a centralized heuristic with complete knowledge as a baseline for design principle, and then followed it by presenting a decentralized solution using similar conventions. Simulation results have shown that MERIT achieved a high content coverage with low start-up delay compared to the state-of-art approach. We envision the usefulness of MERIT as a p2p multi-streaming solution for various interactive multimedia applications.

References

1. Bonald, T., Massoulie, L., Mathieu, F., Perino, D., Twigg, A.: Epidemic live streaming: Optimal performance trade-offs. In: Proc. of SIGMETRICS, pp. 325–336 (2008)
2. Castro, M., Druschel, P., Kermarrec, A.M., Nandi, A., Rowstron, A., Singh, A.: Splitstream: High-bandwidth multicast in co-operative environments. In: Proc. of SOSP, pp. 298–313 (2003)
3. Cheng, W., Liu, D., Ooi, W.T.: Peer-assisted view-dependent progressive mesh streaming. In: Proc. of ACM Multimedia, pp. 441–450 (2009)
4. Cohen, B.: Incentives build robustness in bittorrent. In: Proc. Workshop on Economics of Peer-to-Peer Systems (2003)
5. Cui, Y., Nahrstedt, K.: Layered peer-to-peer streaming. In: Proc of ACM NOSSDAV, pp. 162–171 (2003)
6. Hu, S.Y., Huang, T.H., Chang, S.C., Sung, W.L., Chen, B.Y.: Flod: A framework for peer-to-peer 3d streaming. In: Proc. of IEEE INFOCOM, pp. 1373–1381 (2008)
7. Magharei, N., Rejaie, R.: Prime: Peer-to-peer receiver driven mesh-based streaming. In: Proc. of IEEE INFOCOM, pp. 1415–1423 (2007)
8. Medina, A., Lakhina, A., Matta, I., Byers, J.: Brite: An approach to universal topology generation. In: Proc. of IEEE MASCOTS (2001)
9. Padmanabhan, V.N., Wang, H.J., Chou, P.A., Sripanidkulchai, K.: Distributing streaming media content using co-operative networking. In: Proc. of ACM NOSSDAV, pp. 177–186 (2002)
10. Ren, D., Li, Y.T.H., Chan, S.H.G.: On reducing mesh delay for peer-to-peer live streaming. In: Proc. of IEEE INFOCOM, pp. 1058–1066 (2008)
11. Small, T., Li, B., Liang, B.: Outreach: Peer-to-peer topology construction towards minimized server bandwidth costs. IEEE Journal on Selected Areas in Communication, 35–45 (2007)
12. Tran, D.A., Hua, K.A., Do, T.T.: Zigzag: An efficient peer-to-peer scheme for media streaming. In: Proc of IEEE INFOCOM, pp. 2102–2111 (2003)
13. Xiao, X., Shi, Y., Gao, Y., Zhang, Q.: Layerp2p: A new data scheduling approach for layered streaming in heterogeneous networks. In: Proc. of IEEE INFOCOM, pp. 603–611 (2009)
14. Xu, D., Hefeeda, M., Hambrusch, S.E., Bhargava, B.K.: On peer-to-peer media streaming. In: Proc of ICDCS, pp. 363–371 (2002)
15. Zhang, X., Liu, J., Li, B., Yum, T.K.P.: Coolstreaming/donet: A data-driven overlay network for peer-to-peer live media streaming. In: Proc. of IEEE INFOCOM, pp. 2102–2111 (2005)