Overlay Monitoring and Repair in Swarm-based Peer-to-Peer Streaming [*]

Nazanin Magharei
Department of Computer & Information Science
University of Oregon
Eugene, OR, 97403
nazanin@cs.uoregon.edu

Reza Rejaie
Department of Computer & Information Science
University of Oregon
Eugene, OR, 97403
reza@cs.uoregon.edu

ABSTRACT

In Swarm-based Peer-to-Peer Streaming (SPS) mechanisms, participating peers form a randomly connected mesh over which they incorporate swarm-like content delivery. In practice, a subset of participating peers may form clusters in the overlay due to various reasons such as localization of connectivity within edge ISPs. Despite the commonly held assumptions, the appearance of such clusters could significantly degrade the delivered quality to participating peers in SPS mechanisms.

This paper examines the effect of overlay clustering on the performance of SPS mechanisms for live content. Leveraging the notion of two-phase content delivery in SPS mechanisms, we illustrate the effect of overlay clustering on content delivery. We propose the Overlay Monitoring and Repair (OMR) mechanism as a distributed and scalable approach to maintain proper overlay connectivity in SPS mechanisms. The key idea is to use delivered quality to individual peers as an indication of poor connectivity from other regions of the overlay. OMR employs a probabilistic approach to ensure an adequate number of properly-positioned peers reacts to detected clustering in the overlay without any coordination. Reacting peers rewire a small number of carefully-selected connections in the overlay to significantly improve the performance of content delivery. Our preliminary evaluations demonstrate that OMR mechanism can achieve its goals.

Categories and Subject Descriptors

H.4.3 [**Information Systems Applications**]: Communications Applications—*Internet*

General Terms

Design

1. INTRODUCTION

Swarm-based (or Mesh-based) Peer-to-Peer Streaming (SPS) mechanisms (*e.g.*, [5, 8]) have become increasingly popular for scalable delivery of live content over the Internet (*i.e.*, IPTV-like applications) mainly due to their ability to gracefully accommodate churn

and bandwidth heterogeneity. In SPS mechanisms, participating peers form a randomly connected mesh over which they incorporate swarm-like content delivery. The key component of SPS mechanisms is a pull-based packet scheduling scheme at each peer that determines the pattern of content delivery through the overlay.

The connectivity of the overlay plays a key role on the performance of content delivery in SPS mechanisms. In particular, random nature of connections ensures the diversity of content among connected peers. This in turn enables individual peers to effectively contribute their outgoing bandwidth and leads to the scalability of SPS mechanisms. In practice, a group of peers in the overlay may exhibit a stronger (biased) internal connectivity within the group and weaker connectivity to peers outside the group, *i.e.*, a group of peers form a cluster. Such clusters in the overlay can form for various reasons including the localization of the overlay within an ISP, using regional bootstrap nodes coupled with establishing network-aware connections, and arrival of large number of peers in a short window of time (*i.e.*, flash crowd) . In particular, the localization of overlay connectivity within each edge ISP has received a great deal of attention in recent years [7]. The assumption in these studies is that localization does not adversely affect the performance of P2P applications. Thus, the main focus on these efforts is on providing an interface between ISP and P2P applications to facilitate the localization of connectivity within edge ISPs. In short, the effect of overlay localization on the performance of SPS of live content has not received much attention and is not well understood.

This paper presents our ongoing work on understanding the effect of overlay clustering (or localization) on the performance of SPS mechanisms especially for delivery of live content. In Section 2, we use the notion of "organized view" of a randomly connected overlay to describe two phases of *diffusion* and *swarming* in the content delivery of a certain class of SPS solutions. The notion of two-phase content delivery clearly demonstrates the impact of overlay connectivity on the performance of content delivery. In Section 2.1, using the above insight, we describe how overlay clustering could disrupt the flow of content delivery through the overlay. Furthermore, we quantify the impact of overlay clustering on the performance of content delivery.

The main contribution of this paper is a distributed Overlay Monitoring and Repair (OMR) mechanism that maintains proper connectivity of the overlay. As we discuss in Section 3, the key idea in OMR, is to use delivered quality to individual peers to identify and properly repair any major clustering in the overlay. More specifically, individual peers leverage the unavailability of a substream k as a signal for poor connectivity from subtree k. OMR employs a probabilistic approach to control the number of reacting peers and to increase the likelihood of reaction by properly-positioned peers without any coordination among participating peers. React-
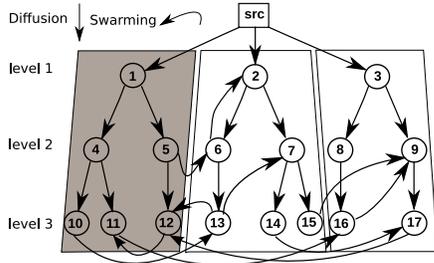
Figure 1: Organized view of a random mesh

ing peers rewire the overlay by swapping their least useful parent with a parent in the subtree with poor connectivity. Our preliminary simulations presented in Section 4, show that OMR can rapidly improve the connectivity of the overlay. Interestingly, even the resulting rewired overlay by OMR mechanism, is significantly more localized than a random overlay mesh. This suggests that there is an opportunity for overlay localization without compromising the performance of SPS of live content.

2. BACKGROUND

This section provides a brief overview of the SPS of live content which has two basic components of *Overlay Construction* and Content Delivery.

Overlay Construction: Peers maintain a directed and randomly connected mesh where each peer has several parents and children. The number of parents and children that each peer tries to maintain is proportional with its incoming and outgoing bandwidth, respectively. In other words, all peers try to maintain the same ratio of incoming (outgoing) bandwidth to incoming (outgoing) degree (*i.e.*, bandwidth-degree ratio) which is a configuration parameter. This condition ensures that all connections have roughly the same average bandwidth and avoids any bottleneck at the incoming (outgoing) access link of individual peers.

Content Delivery: Peers perform swarming content delivery which combines push content reporting by parents with pull content requesting by children. Each peer simultaneously receives content from all of its parents and provides content to all of its children. Each peer, as a parent, progressively reports the availability of its new packets to all of its children. Given the available packets at individual parents, a packet scheduling scheme at each peer periodically determines a list of packets that should be requested from each parent. Each parents simply deliver requested packets by each child in the provided order and at the rate that is determined by the congestion control mechanism.

To demonstrate the effect of overlay connectivity on the performance of SPS, we use an organized view of a randomly connected and directed overlay. Suppose the distance of a peer *p* from the source is determined by its shortest path (in hops) from source through the overlay. Then, peers that have the same distance of *n* hops from the source can be grouped into level *n*, as shown in Figure 1. Given the average outgoing peer degree of ($OutDeg$), source degree of Deg_{src} and the total peer population of N, the number of peers at level *n* is limited to ($Deg_{src}*OutDeg^{(n-1)}$), and the number of levels (Max_Depth) is limited to $Max\_Depth \geq \log_{OutDeg}(N/Deg_{src})$.

We consider a resource constraint setting where *(i)* source has limited bandwidth which is sufficient to send a live video stream without any redundancy (*i.e.*, a single copy of each packet), and *(ii)* the demand for resources (aggregate incoming bandwidth) is equal to the available resources (aggregate outgoing bandwidth), *i.e.*, resource index is 1. We focus on a class of SPS mechanisms that employs a packet scheduling scheme which prioritizes pulling

of packets with the largest timestamp [1, 5]. This class of packet scheduling schemes exhibits a significantly better performance than others (*i.e.*, rarest or random) [1, 2, 4] by maximizing the utilization of outgoing bandwidth among participating peers. The pattern of delivery for a single packet over an organized overlay in such a packet scheduling scheme consists of the following two phases: *(i) Diffusion phase* where a packet is pulled from a peer in level *n* by all of its children in level *n* + 1. All connections from peers in level *n* to peers in level *n* + 1 ($n<Max\_Depth$) are used for diffusing new packets through the overlay and thus called *diffusion connections*. The diffusion connections are shown with straight edges in Figure 1. Since the source can only send a single copy of each packet (due to its limited bandwidth), the collection of peers that receive a packet during its diffusion phase form a subtree that is rooted at a peer in level 1. We call this a diffusion subtree and Figure 1 shows the three "diffusion subtrees" rooted at peer 1, 2 and 3 in an overlay. *(ii) Swarming Phase* where a packet is pulled from a peer in level *n* by its children in the same or lower levels ($m \leq n$) of the overlay. All the connections from a peer in level *n* to a peer in level *m* ($\leq n$) are used for swarming and thus called *swarming connections*. These connections are shown with curly edges in Figure 1. In essence, swarming connections are used for the delivery of a packet from peers in the packet's diffusion subtree to other peers in the overlay.

The organized view of the overlay allows us to clearly separate the diffusion and swarming connections based on the relative position of related peers in the overlay. The collection of packets that are delivered to a particular peer in level 1 and thus diffuse through the same diffusion subtree are called a *substr* of the content. Since each packet is sent only once, the content of different substreams are mutually exclusive. We identify these substreams by the id of the root of their corresponding subtree. For example, in Figure 1, the shaded subtree rooted at peer 1 has the $substr_1$ while the other subtrees rooted at peer 2 and 3 deliver $substr_2$ and $substr_3$, respectively.

2.1 Impact of Overlay Connectivity

The notion of diffusion and swarming phases for delivery of each substream (or packet) enable us to relate buffer requirement and delivered quality for individual peers.The maximum number of required intervals for the diffusion of each substream is equal to the Max_Depth of its diffusion subtree. The number of swarming intervals for a substream to a peer depends on the relative location of its swarming parents. Typically, a peer has one diffusion parent and several swarming parents. If the swarming parent of a peer is located in the diffusion subtree of the corresponding substream, then it only needs one extra interval to receive that substream. In general case, different swarming parents may not be located at the proper diffusion subtrees and thus require up to three swarming intervals in a randomly connected overlay which is derived mathematically and through simulation [5, 8]. For example, peer 7 in Figure 1, has the parent 13 in the same diffusion subtree. In this case, peer 7 can receive $substr_1$ through a longer path from parent 13. The buffer requirement at each peer should be set based on longest number of swarming intervals and therefore is set to Max_Depth+3.

Swarming connections can be divided into four categories based on the relative location of connected peers [6] as follows: (i) C_{ld}: connecting peers at the bottom of two different diffusion subtrees (*e.g.*, connection from 10 to 13 in Figure 1), (ii) C_{ls}: connecting peers at the bottom of the same diffusion subtree (*e.g.*, connection from 12 to 11 in Figure 1), (iii) C_{id}: connecting a peer at one diffusion subtree (bottom or not) to an internal peer at a different diffusion subtree (*e.g.*, connection from 15 to 9 or 5 to 6), and (iv) C_{is}:

connecting a peer at one diffusion subtree (bottom or not) to an internal peer at the same diffusion subtree (*e.g.*, connection from 13 to 7 or 6 to 2). Swarming connections that are between peers in different diffusion subtrees (*i.e.*, C_{ld} and C_{id}) are more useful for effective delivery of content. In particular, C_{ld} connections are most useful for swarming. These connections provide $substr_k$ for peer p at the bottom of diffusion subtree l. Since p is at the bottom of its own diffusion subtree (*i.e.*, its outgoing connections are not used for diffusion), it can effectively relay $substr_k$ to other peers in the same or different diffusion subtrees. Therefore, increasing C_{ld} swarming connections results in a large improvement in delivery of all substreams to individual peers within a small buffer size.

2.2 Effect of Overlay Clustering

In practice a group of peers may have a higher tendency to connect to each other and form clusters. Such a clustering effect might occur for various reasons. For example, ISP may provide an interface (such as P4P [7]) for its client peers to discover and connect to each other in order to reduce the number of external connections and thus limit the associated traffic. Peer discovery mechanisms that rely on local bootstrapping node and ping response from discovered peers are likely to exhibit similar clustering effect among close-by peers. Furthermore, a combination of flash-crowd event with certain pattern of peer arrival could also lead to cluster formation. We note that the dynamics of peer participation may reduce such a clustering event in some but not all scenarios.

Formation of clusters in the overlay causes a poor connectivity between different clusters that limits the flow of content among them. This may adversely affect the performance of content delivery. For example, if incoming external connections for a cluster do not pull mutually exclusive substreams, the full quality version of the stream does not reach that cluster and thus all peers in the cluster receive lower quality stream.

To further elaborate on the effect of overlay clustering on content delivery, we note that the diffusion subtrees and swarming connections in a clustered overlay are formed based on the relative distance of peers from the source as we described earlier. Peers in a particular cluster can be mapped to one or multiple diffusion subtrees depending on their external connections to other clusters and source. For example, if all peers in a cluster become part of a diffusion subtree, then the subtree should have many swarming connections within the subtree (of type C_{ls} or C_{is}), and very few swarming connections to other subtrees (of type C_{ld} or C_{id}). The limited number of swarming connections from $subtree_k$ to other subtrees could increase the number of required swarming intervals to receive $substr_k$ or could make $substr_k$ unreachable to other peers even with a large number of swarming intervals.

Alternatively, peers in a cluster could be mapped to different diffusion subtrees. In this case, the effect of biased connectivity on content delivery may vary depending on the relative location of peers in different subtrees. The swarming connections within a cluster connect corresponding regions of two subtrees at the cost of lower (or no) swarming connections to other diffusion subtrees. Such a focused inter-subtree swarming connections are not very useful because they may connect peers at the higher levels of two diffusion subtrees. In essence, any clustering in the overlay, decreases the number of C_{ld} connections between subtrees compared to the random overlay which adversely affects the performance of content delivery.

We use simulations to demonstrate the effect of overlay clustering on the content delivery. Consider 5000 homogeneous peers with $InDeg$ and $OutDeg$ of 12, that are grouped into 10 clusters (or ISPs). Source has sufficient bandwidth to deliver a single copy
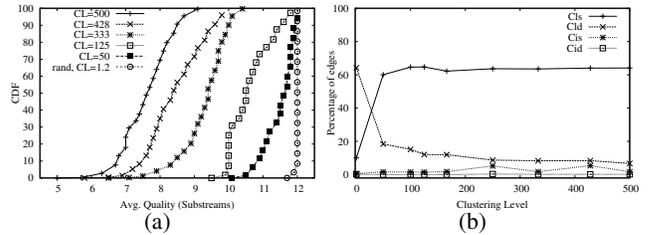
Figure 2: Effect of clustering on content delivery: (a) Distribution of delivered quality. (b) Percentage of various swarming connections.

of each packet (or substream), therefore, its $OutDeg$ is 12 which implied that the number of subtrees are 12. Figure 2(a) depicts the distribution of delivered quality to individual peers for various level of clustering. We define the level of clustering CL based on the total number of incoming connections to all peers in a cluster divided by the number of external incoming connections to the cluster. Figure 2(a) shows that the delivered quality significantly decreases with the level of clustering. Even with a relatively low level of clustering (*e.g.*, 50) the performance is smaller than the random overlay which has the clustering level of 1.2. Figure 2(b) depicts the percentage of various types of swarming connections as discussed in Section 2.1, as a function of the level of clustering. This figure reveals that with a higher level of clustering, C_{ld} connections decrease while C_{ls} connections increase. On the other hand, when we move towards a random overlay the number of good connections between subtrees (*i.e.*, C_{ld}) increases and those undesired connections within a subtree (*i.e.*, C_{ls}) decrease.

In summary, despite the subtle effect of clustering on overlay connectivity, its primarily relevant impact on content delivery is the limited (or no) availability of subtstream(s) that are associated with subtrees with limited swarming connectivity. Furthermore, the higher the number and quality (*i.e.*, type) of swarming connections from subtree k to subtree l are, the larger the number of peers in subtree l that can not receive $substr_k$ would be. In the next section, we leverage this point to devise a mechanism to detect poor connectivity in the overlay.

3. OVERLAY MONITORING & REPAIR

We propose a QoS mechanism, called OMR, to maintain the connectivity of the overlay such that swarming content delivery operates properly, and a large fraction of peers receives the desired quality. Such a QoS mechanism, requires a *detection* component that identifies any problem with overlay connectivity, and a *reaction* component that repairs the overlay connectivity by rewiring the minimum number of connections. The QoS mechanism should not require coordination among peers and should be light weight in order to scale to large groups. It should also minimize the number of changes in the connectivity of the overlay to limit the resulting dynamics in the overlay.

The main intuition in OMR is as follows: when a good packet scheduling scheme is used, the limited availability of $substr_k$ in the diffusion subtree l indicates poor connectivity from subtree k to subtree l. Therefore, one can monitor the performance of content delivery (*i.e.*, delivered quality to individual peers) instead of connectivity in the overlay. This is an important distinction for two reasons: *(i)* identifying problems with overlay connectivity in a scalable fashion is expensive, and *(ii)* the effect of overlay connectivity on content delivery is rather subtle. We argue that the performance of content delivery is the only relevant metric since any type of clustering that does not affect content delivery, is not a concern.
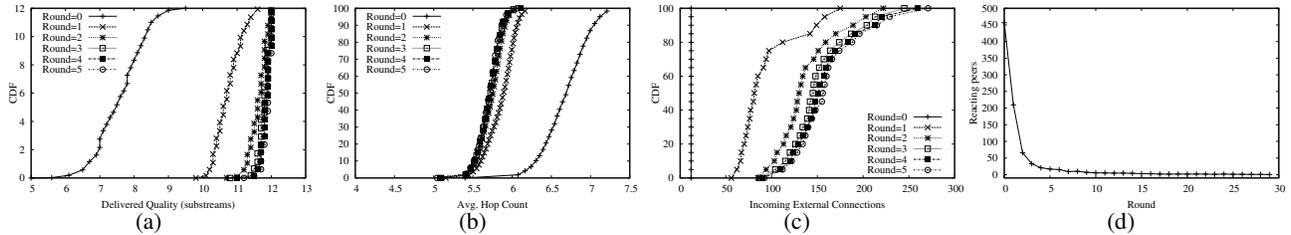
Figure 3: OMR basic behavior: (a) Distribution of delivered quality. (b) Distribution of average hop count across peers in various rounds. (c) Distribution of incoming external connections to ISPs. (d) Percentage of reacting peers.

Our proposed OMR mechanism is distributed and demand-driven by only leveraging the observed quality at each peer. Each peer p monitors its received substreams (*i.e.*, received quality) as well as the available substreams among its swarming parents (*i.e.*, available quality). When Peer p is missing at least one substream for a given period of time, it periodically invokes a *Reaction Algorithm* to determine whether it should react. If peer p is selected to react, it invokes a *Repair Algorithm* that implements the minimum number of changes in overlay connections to achieve the maximum improvement in the delivered quality.

We assume that each packet carries a hop count that is set to zero and increases by each node that relays the packet forward. This allows each peer to determine its shortest path from source (and its diffusion parent) as the overlay evolves. Source delivers only a single copy of each packet and it tags individual packets by their substream id based on the child peer (*i.e.*, root of the diffusion subtree) where the only copy of the packet is delivered. The substream tag in packets allows each peer to estimate the delivery of each substream and its availability among its swarming parents. Finally, each peer is aware of "the effective (or average) Max_Depth" of the overlay. This information can either be deducted from the hop count of received packets or be estimated from the population and degree of participating peers.

Reaction Algorithm: A problem in overlay connectivity often affects delivered quality to a number of peers. The reaction algorithm is independently run by individual peers to determine which subset of affected peers should react to a drop in quality. More specifically, this algorithm tries to achieve two goals: *(i)* limiting the number of reacting peers, and *(ii)* increasing the probability of reaction by peers that are able to achieve the most improvement in overlay connectivity for a fixed number of rewiring operations due to their position in the overlay. The basic problem is very similar to selecting proper nodes to send a repair request in Scalable Reliable Multicast (SRM) [3]. Thus, we adopt a similar "probabilistic approach" to control the number and the location of reacting peers. Each affected peer periodically estimates the probability of reacting to a missing substream. The reaction probability for peer p to the absence of $substr_s$ depends on the following factors:

- *Contribution Factor (cf)*: the fraction of p's swarming children that does not receive $substr_s$. If p receives $substr_s$, it can relay it to all these swarming children that are missing s.

- *Parent Factor (pf)*: the fraction of p's swarming parents that are located in the same diffusion subtree and do not receive $substr_s$. These parents are experiencing the same problem and may react as well. Note that the swarming connection from these parents are of type C_{ls} and C_{is} that are less useful for content delivery.

- *Depth Factor (df)*: the relative value of p's depth to the overlay depth (Max_Depth). This is motivated by the fact that the outgoing connections of peers at lower levels (*i.e.*, higher relative depth) are mostly swarming connections.

The probability of reaction is biased towards peers with larger po-

tential contribution, smaller parent factor and larger relative depth. A peer with the above qualities would be perfectly positioned to repair the overlay. The reaction probability is computed as $cf^{\alpha} * df^{\beta} * \frac{1}{pf^{\gamma}}$ where α, β and γ determine the contribution of the above factors on the responsiveness of OMR.

Probability of reaction for a peer p to subtree deficit of s, is proportional to the contribution of the peer. More specifically, if p has a large number of swarming children that do not have s, it is more probable to react. On the other hand, if p has a large number of swarming parents in its own diffusion subtree that do not receive s, the probability of reaction for p should be decreased as those parents are likely to react. Moreover, the lower the depth of the peer, the higher the probability of reaction would be as peers at the bottom of the diffusion subtree could contribute the swarming content.

Repair Algorithm: Once the reaction algorithm triggers a reaction, it should identify a proper position nd, in the missing subtree s. The goal is to place a peer with higher contribution at a higher position in the missing subtree s. The intuition behind this design choice is that when peer p has many swarming children which are missing subtree s, peer p will be their parent for subtree s. Thus to shorten the buffer size for peers, it is desirable to position peer p at a higher level in the diffusion subtree s.

To prevent from selecting the same depth by multiple reacting peers, we compute the new depth probabilistically. Towards this end, we form a probability function: $P(nd, cf) = 1 - (1 - cf)^{(nd-1)}$. The value of the function increases by both cf and nd. We start from potential new depth of 2 and at each step we decide to choose the value ND as the new depth with probability $P(ND, CF)$. We repeat this process until we decide on an ND or we reach $ND = Max\_Depth_s$.

Upon computing the position in the $subtree_s$, peer p traverses the subtree in a depth-first search fashion by selecting a random child of each peer till it reaches the corresponding level. Finally, the reacting peer switches its current parent who does not provide any substream with this new parent. In case the new parent does not have an empty slot to admit a new child, it will disconnect one of its current children which has the least useful connection type (*i.e.*, of type C_{ls} and C_{is}). This new disconnected child will be swapped by peer p's previous parent.

4. PERFORMANCE EVALUATION

In this section, we present the preliminary evaluation of OMR using our event-driven simulator called *psim*. The simulation is performed in multiple rounds where each round includes two steps: *(i)* P2P content delivery over the existing overlay and *(ii)* running the OMR mechanism at each peer which may cause rewiring of some connections. The buffer size that is set to $\log_{OutDeg}(N/Deg_{src})$+3. The delivered quality to individual peers is measured based on the number of delivered substreams. Clearly, if the connectivity of the overlay is inadequate, some peers may not receive all the required substreams and thus observe lower delivered quality. Round 0 starts
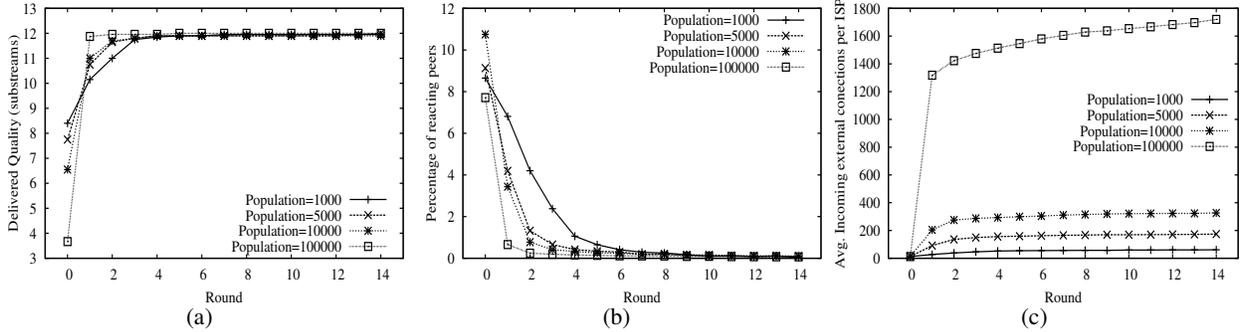
Figure 4: Scalability of OMR: (a) Distribution of delivered quality. (b) Percentage of reacting peers. (c) Average number of external incoming connections per ISPs.

with a "localized overlay" and the overlay is progressively rewired as a result of running OMR mechanism across individual peers in each round. In essence, each round can be viewed as the interval between two consecutive repair events. The simulation ends when the overlay does not experience any further change for a few rounds. For simplicity, we do not incorporate churn in our simulations.

We use the following default settings in our simulations: The required incoming degree to receive full quality stream is 12. The source bandwidth is equal to the stream rate, therefore source degree is also 12. Peers have homogeneous and symmetric bandwidth. Thus all peers are able to receive full quality stream and resource index is 1. Peer population is 5000 that is divided into 10 equal-sized clusters (or ISPs). The initial "localized overlay" has the highest level of clustering where each cluster has the minimum number of external connections, namely 12, and clusters form a randomly connected mesh. α, β and γ are set to 1, 2 and 2, respectively. The reported results for each simulation are averaged across ten runs with different random seeds.

4.1 Basic Characteristics of OMR

Each line of Figure 3(a) depicts the CDF of delivered quality in separate rounds in our default scenario. In essence, this figure illustrates the incremental effect of OMR rewiring in each round on the delivered quality. This figure reveals that OMR leads to significant improvement in quality that occurs mostly during the first two rounds. Each line in Figure 3(b), shows the CDF of the average hop count for delivered packets to individual peers in different rounds. This figure clearly demonstrates how OMR effectively tightens the connectivity and rapidly reduces average hop-count for delivery of packets to all peers in the overlay within a couple of rounds. The distribution of maximum hop count shows a similar trend and is not presented due to the limited space.

Figure 3(c) illustrates the effect of OMR on the increasing connectivity among clusters by showing the distribution of cumulative number of external connections across all clusters (ISPs) during the initial rounds. In round 0, all clusters in the localized overlay have exactly 12 external edges. After one round, OMR results in 50 to 170 external connections for different clusters. After two rounds, the number of external connections increases to the range 90 to 220. The number of external connections quickly stabilizes and does not exhibit any significant increase during the following rounds. In fact, no more rewiring occurs after the 10th round. The relative diversity in the number of established external connections among clusters caused by their initial external connectivity and relative position from the source. For example, a cluster that has two incoming connections from source contains two diffusion subtrees and establishes fewer external connections. It is worth noting that the resulting rewired overlay is still much more clustered than a comparable random overlay.

Finally, Figure 3(d) depicts the number of reacting peers (or the number of rewired operations) in each round. This figure reveals that roughly 9% of edges (*i.e.*, 450 edges) are rewired in the first round. However, the number of rewired edges rapidly drops in the following rounds. This figure shows that OMR achieves a short response time by aggressively rewiring the overlay during a few rounds at the cost of rewiring a relatively large percentage (9%) of edges in a single round. Since the number of rewired edges in one round could be a concern, the reaction algorithm can be tuned through the configuration parameters α, β and γ introduced in Section 3, to reduce the number of reacting peers. This controls the number of rewiring events in one round but is likely to increase the response time. We are currently exploring this issue.

4.2 Scalability of OMR

To examine the scalability of the OMR, we evaluate its behavior on overlays with 1K, 5K, 10K and 100K peers. Each line in Figure 4(a) shows the average delivered quality to peers in an overlay with certain population as a function of time (rounds). This figure reveals that the average delivered quality rapidly improves during the first couple of rounds and reaches the full quality regardless of peer population. Figure 4(a) also reveals that the delivered quality in large overlays is initially lower than smaller overlays. This is due to their lower ratio of external to internal connections for each cluster in larger overlays. Note that the number of external connections for the localized overlay is fixed at 12. To generate larger localized overlays, we increase the number of peers and edges within each clusters without increasing the number of external connections. This in turn increases the level of clustering in the overlay and adversely affects the performance of content delivery. Figure 4(a) indicates that the time to reach the maximum quality is shorter in larger overlays despite their higher quality deficit in round zero. The reason is that the lower delivered quality in larger overlays triggers a larger number of peers to react initially which leads to a rather larger improvement in the delivered quality.

To quantify the reaction of OMR mechanism with a different population of peers, Figure 4(b) depicts the percentage of reacting peers as a function of rounds for overlays with different populations. This figure shows that roughly the same fraction of peers react in the first round regardless of peer population. However, the absolute number of reacting peers in a larger overlay is proportionally larger. After the significant increase in the delivered quality the first round, the percentage of reacting peers in the following rounds rapidly drops to zero for all overlays with any population.

To characterize the effect of rewiring on the connectivity among ISPs, Figure 4(c) shows the average number of external incoming connections per ISP as a function of rounds. This figure shows that the number of external connections rapidly increases to a certain level (*i.e.*, 80, 200, 350, and 1700) and further stabilizes. We note
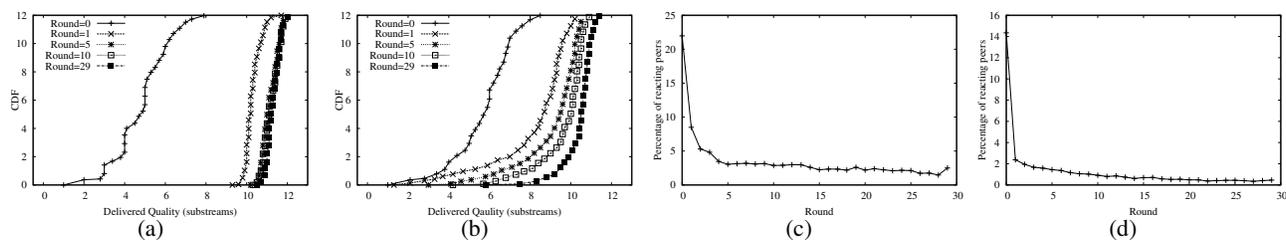
Figure 5: Effectiveness of OMR in presence of BW heterogeneity: (a) and (b) are the distribution of delivered quality across high-bandwidth peers in $SC1$ and $SC2$, respectively. (c) and (d) are the percentage of reacting peers in $SC1$ and $SC2$, respectively.

that the swarming intervals in a clustered overlay could be larger than 3 intervals due to the limited external connectivity among clusters. The higher the level of clustering, the larger the required number of swarming intervals. Since we fix the number of swarming intervals in the simulations, a larger number of peers receive a low quality in larger overlays which triggers the repair by a larger number of peers. These repairs result in establishing new external connections between clusters. The external connections serve as shortcuts in the overlay, reduce the number of swarming intervals, and thus increase the delivered quality which prevents further reaction by other peers. As part of our future work, we plan to mathematically derive the minimum required external connections for each cluster to ensure high delivered quality to individual peers in various scenarios. This reveals any potential gap between the required and the observed external connectivity in these scenarios.

4.3 Peer Bandwidth Heterogeneity

We now examine the performance of OMR mechanism in more realistic scenarios with two groups of peers with high and low symmetric access link bandwidths. We assume that content is encoded with MDC and high bandwidth peers are able to receive the full stream quality (all 12 substreams) while low bandwidth peers can only receive 50% of the quality (6 substreams). The incoming and outgoing degree of high and low bandwidth peers are 12 and 6, respectively. We consider two scenarios of $SC1$ and $SC2$ with 5000 peers where 80% and 20% of peers are high bandwidth and all other peers are low bandwidth, respectively. Low bandwidth peers require a smaller number of substreams and are less sensitive to unavailability of a particular substream. Therefore, we primarily focus on the delivered quality to high bandwidth peers.

Figures 5(a) and 5(b) depict the distribution of delivered quality to high bandwidth peers in different rounds of applying OMR mechanism to both scenarios. The distribution of delivered quality in both scenarios are rather similar in round 0, and OMR mechanism result in significant improvement in delivered quality during the first two rounds. However, the distribution of the delivered quality in $SC1$ (with 80% high bandwidth peers) quickly becomes uniform after a couple of rounds, while it always remains relatively skewed in $SC2$ with lower median for delivered quality. The observed behavior in $SC2$ can be explained as follows: the large percentage of low bandwidth peers in $SC2$ increases the likelihood that a high bandwidth peer p is connected to several low bandwidth peers as a parent or child in the overlay. Assuming peer p misses a substream s and some of its low bandwidth parents/children reach their target quality and do not require substream s. This leads to the decrease of the value of contribution and parent factors in the reaction algorithm for the high bandwidth peer p. Thus, the probability of reaction by some high bandwidth peers decreases despite the deficit in their quality in $SC2$.

Figures 5(c) and 5(d) depict the percentage of reacting peers (or rewiring events) in the above two scenarios. We observe that the response time in these heterogeneous scenarios is longer than the

homogeneous scenario examined in earlier subsections. This is due to the subtle interaction of two factors *(i)* the number of low bandwidth parent/children for each high bandwidth peer, and *(ii)* whether these low bandwidth peers require the same missing substream or not. Figures 5(c) reveals that the number of reacting peers appears to stabilize at a low rate but does not reach zero. This persistent residual reaction occurs when a particular rewiring event replaces one low bandwidth parent with another low bandwidth parent. This could ensure availability of a particular substream at the cost of losing another substream. As part of our future work, we plan to examine the above issues that arise in the behavior of OMR over heterogeneous overlays.

5. CONCLUSION AND FUTURE WORK

In this paper, we presented an overlay monitoring and repair (OMR) mechanism for swarm-based P2P streaming schemes. OMR leverages the availability of content at individual peers to detect poor connectivity between different regions of the overlay due to regional clustering. OMR uses a probabilistic approach to ensure an adequate number of affected peers with proper position in the overlay react to poor connectivity. Reacting peers carefully identify proper counterparts in the overlay and swap specific parents to improve the connectivity of the overlay with the minimum number of rewiring operation. Our preliminary simulations results demonstrated that OMR is able to achieve its design goal, and revealed a few more issues that require further investigation. As part of our future work, we plan to investigate the following issues on the behavior of OMR mechanism in more detail: *(i)* the effect of churn and peer bandwidth heterogeneity, *(ii)* the effect of reaction algorithm and its damping factor on controlling the tradeoff between responsiveness and aggressiveness of OMR mechanism.

6. REFERENCES

[1] T. Bonald, L. Massoulie, F. Mathieu, D. Perino, and A. Twigg. Epidemic live streaming: Optimal performance trade-offs. In *SIGMETRICS*, 2008.

[2] C. Liang, Y. Guo, and Y. Liu. Is random scheduling sufficient in P2P video streaming? In *ICDCS*, 2008.

[3] S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang. Reliable Multicast Framework for Light-Weight Sessions and Application Level Framing. *ACM/IEEE Transactions on Networking*, 1997.

[4] N. Magharei and R. Rejaie. Dissecting the performance of Live Mesh-based P2P Streaming. Technical Report CIS-TR-07-05, 2007.

[5] N. Magharei and R. Rejaie. PRIME: Peer-to-Peer Receiver-drIven MEsh-based Streaming. *ACM/IEEE Transactions on Networking*, 2009.

[6] N. Magharei, R. Rejaie, and Y. Guo. Mesh or Multiple-Tree: A Comparative Study of P2P Live Streaming Services. In *INFOCOM*, 2007.

[7] H. Xie, R. Yang, A. Krishnamurthy, Y. Liu, and A. Silberschatz. P4P: Provider portal for P2P applications. In *ACM SIGCOMM*, 2008.

[8] X. Zhang, J. Liu, B. Li, and T. Yum. Coolstreaming: A data-driven overlay network for live media streaming. In *INFOCOM*, 2005.