Characterizing Locality-aware P2P Streaming

***Abstract*—Peer-to-Peer (P2P) streaming systems have been increasingly popular and successful in today's Internet, which provide large numbers of videos to millions of users at low server costs. While P2P streaming makes the best use of peers' upload bandwidth to alleviate the server load, they incur larger volumes of inter-ISP traffic, which significantly threatens the benefits of Internet service providers (ISPs). There have recently emerged proposals advocating locality-aware P2P streaming protocol design, which aim to minimize the amount of inter-ISP traffic. Nevertheless, theoretical analysis and in-depth understanding of the impact of such design on the performance of P2P streaming are still missing, which once done, can provide useful insights on the design of P2P streaming systems, towards a desirable tradeoff between traffic locality and streaming performance. In this paper, we discuss our research on performance modeling of locality-aware P2P streaming. We characterize the relationship between streaming performance and traffic locality, starting from a basic network with two ISPs and homogeneous peer bandwidths, to more complicated and practical networks with multiple ISPs and heterogeneous peer bandwidth distribution. We also seek to evaluate our models and theoretical results with large-scale simulations under realistic settings.**

I. Introduction

Peer-to-Peer overlay network helps participating peers contribute their upload capacities and other resources(such as CPU, storage). This alleviates the burden of servers and makes applications scalable. So, P2P applications are becoming more and more popular. Live peer-to-peer streaming has gained growth on the Internet. Unlike the client-server streaming, P2P streaming can reduce the bandwidth requirement of servers, this will reduce the cost as the viewers increase greatly nowadays and make the streaming scalable.

Much research has been done on exploring the best performance of live streaming through peer-to-peer network. A large number of P2P streaming protocols have been proposed. They generally can be categorized into two types: Push-based tree streaming strategies and Pull-based mesh streaming strategies. Push-based tree streaming organizes participating peers into one or more multicast trees and when the parent node has received new chunks, it will push them to its children. So, a peer don't need to send requests for chunks needed. Without the delay of requests, the chunk dissemination is usually faster. However, the tree structure is not robust to peer dynamics. When there are peers leaving or arriving, the cost of sustaining the trees are high. In contrast, Pull-based mesh streaming strategy is robust to peer dynamics, so, real-world systems adopt pull-based mesh streaming strategy to accommodate to peer dynamics in real world.

As the pull-based mesh streaming strategy prevails, much work has been done on finding the performance bound of pull-based mesh streaming strategy. a lot of theoretical work tries to give the streaming capacity. Yong Liu has proposed the snowball algorithm of streaming to achieve the minimum delay of chunk dissemination. Minghua Chen has studied the streaming capacity with node degree constraints.

However, no theoretical work has considered the inter-ISP traffic brought by P2P streaming when trying to achieve the best performance. P2P network can alleviates the cost of servers and make the applications scalable. At the same time, P2P network increases the inter-ISP traffic, which increases the cost of ISPs. So, P2P has moved the cost from servers to ISPs. This has already drawn the attention of ISPs. So, traffic locality of P2P applications has recently become important. Much work has been done on P2P file sharing. But the work on traffic locality of P2P streaming is little. In this paper, we propose an inter-ISP traffic model, Inter-Peer model, to analyze how the traffic locality will affect the streaming performance.

II. Related work

As P2P streaming becomes popular and successful in distribution of Internet video, a lot of work has been done on analyzing the fundamental properties and performance bounds of P2P streaming. Zhou et al. [] developed a simple model for streaming systems using pull-based mesh protocols. They use this model to evaluate and compare different chunk selection strategies. Their work provides a good guide on the design of chunk selection algorithm. Bonald et al. [] studied the streaming rate and minimum delay of several push-based protocols using specific peer selection strategy and chunk selection strategy and figured out which ones can achieve near-optimal rate and delay. Zhou et al. and Bonald et al.'s work is intended to achieve a better performance in P2P streaming. In [1] and [], the maximum sustainable streaming rate and minimum delay bound of pull-based mesh protocols have been given respectively. Liu et al. [] has given the performance bounds for push-based streaming protocols. Feng et al. [] figured out the performance gap between pull-based mesh streaming protocols and fundamental limits due to periodic buffer map exchanges and decentralized algorithm. All the work investigates streaming properties and performance without considering inter-ISP traffic problem.

The increasing inter-ISP traffic due to P2P streaming video is drawing the attention of ISPs. To avoid the filtering and blocking of ISPs, ISP-friendly P2P streaming is proposed, in which traffic locality-awareness is considered. P4P [2] presents an architecture by which ISPs provide P2P applications necessary information for them to make peer selection decisions, which take both performance and traffic into considerations. Fabio Picconi et al. [3] proposes a two-overlay mesh protocol to achieve the traffic locality and ensure the performance of

streaming. The primary overlay is highly-clustered to reduce inter-ISP traffic as much as possible. The secondary overlay creates links between peers in different ISPs. And the links of the secondary overlay are dynamically unchoked to guarantee the performance under dynamic network environment. Nazanin Magharei et al. [] represent each ISP as a single peer and external connections between different ISPs constitute a top-tier overlay. They use inter-ISP scheduling to ensure each ISP receives all substreams of video. Then, intra-ISP scheduling is used to deliver all substreams received by edge peers of an ISP to all internal peers. [2] [3] [] propose a mechanism to realize traffic locality respectively. With respect to commercial streaming systems, [4] measures the traffic of PPLive and shows that PPLive achieves a high ISP level traffic locality. No modeling work on locality-aware P2P streaming has been done yet. Our work models the locality-awareness of P2P streaming. Based on the model, we analyze the streaming performance under traffic locality and figure out the effect of inter-ISP traffic and the necessary inter-ISP traffic to achieve good performance.

III. Problem Motivation and System Model

A. Problem Motivation

The large volumes of inter-ISP traffic induced by ISP-agnostic P2P streaming protocols increase ISPs' cost. Locality-aware P2P streaming is proposed to avoid the P2P packet filtering and blocking by ISPs. The in-depth understanding of locality-aware streaming protocols requires modeling work on them. Our paper models the locality-aware mesh overlays to explore the impact of traffic locality on the streaming performance and the necessary inter-ISP traffic to achieve good performance.

B. System Model

In this section, we present our basic model of locality-aware P2P streaming protocols, including the underlying assumptions and key notations summarized in Table I. The upload capacity of peers is the bottleneck. We separate the peers in an ISP into different classes according to their upload capacities. Let U_{pij} denote the upload capacity of class j peers in ISP i and u_{pij} be the relative peer upload capacity, which is the ratio of U_{pij} to the streaming rate R. If the peers in ISP i have the same upload capacity, we denote the capacity by U_{pi}. The server capacity of ISP i is denoted by U_{si}, the relative server capacity is u_{si}, which is the ratio of U_{si} to the streaming rate R. The number of participating peers in ISP i is N_i. The total number of peers in streaming system is N.

Notation Table

Notation	Definition
M	number of ISPs
N	the system scale
N_i	the peer scale of ISPi
U_s	server capacity
U_{pi}	peer upload capacity of ISPi in homogeneous case
U_{pij}	peer upload capacity of class j peers in ISPi
u_s	relative server capacity
u_{pi}	relative peer upload capacity of ISPi
C	number of partners of a peer.
T_m	the inter-ISP traffic of ISPm.
T_{mi}	the flow-in traffic from other ISPs of ISPm.
T_{mo}	the flow-out traffic to other ISPs of ISPm.

In this paper, we model locality-awareness in mesh streaming protocols. The links between peers in different ISPs are controlled under traffic locality mechanisms. To model the locality mechanisms, we assume there exist imaginary Inter-Peers between peers in different ISPs. These Inter-Peers serve as bridges to build links between peers in different ISPs. And every Inter-Peer controls a link and downloads one copy of streaming chunks from peers in an ISP and serve peers in another ISP. Without considering the details of realization, we assume Inter-Peers know to download the needed chunks from which peer and serve the chunks to which peer that needs them. The number of Inter-Peers downloading chunks from ISPi to ISPj n_{ij} can be used to quantify the inter-ISP traffic. In live streaming, every peer's downloading rate equals to playback rate R. So, for one Inter-Peer, the downloading rate is R. With the number of Inter-Peers and downloading rate R, we can calculate the inter-ISP traffic from ISPi to ISPj is $n_{ij} \cdot R$. Because the focus of our work is on the modeling of traffic locality, the links among peers in the same ISP are not limited.

The controlled inter-ISP links of locality-aware P2P streaming protocols can be well modeled by Inter-Peers. And inter-ISP traffic can be quantified by the number of Inter-Peers. With this model, we are able to analyze the impact of traffic locality on the performance of live streaming. And we use the chunk dissemination delay as the metric for peers' performance, because live streaming video is a delay sensitive application. The shorter the chunk dissemination delay is, the shorter the startup delay is.

IV. Study of impact of traffic locality on performance

We first analyze the homogeneous case. And then we extend to the heterogeneous case.

A. Homogeneous & Balanced Case

In the homogeneous and balanced case, all the peers' upload capacities are the same, denoted by U_p, and we define the relative capacity u_p of peers as the ratio of the upload capacity U_p to the streaming rate R. The server's upload capacity is U_s, and the relative server capacity is u_s. And we denote the number of neighbors of a peer by C. The servers pump out chunks to peers at the first hop, hop 1. And after a peer receives

a chunk at hop i, it transmits the chunk to C neighbors at hop i+1.

First we analyze the ISP-agnostic streaming. We can see how much unnecessary inter-ISP traffic is induced by ISP-agnostic streaming through comparison of ISP-agnostic streaming and Locality-aware streaming.

Lemma 1: For a streaming system with server relative upload capacity u_s, and each peer with C neighbors, the maximum number of hops needed for all peers to get a chunk is:

$$H_{max} = 1 + \lceil log_{(1+C)} \frac{N}{u_s} \rceil$$

Where N is the number of peers in the system.

The Maximum No. of peers with a chunk after i hops is:

$$N(i) = min\{N, u_s(1+C)^{i-1}\}$$

So, No. of hops for all peers getting a chunk is:

$$u_s(1+C)^{i-1} \geq N \quad (1)$$

$$H_{max} = 1 + \lceil log_{(1+C)} \frac{N}{u_s} \rceil \quad (2)$$

We assume the time for playing one chunk is one time unit. The servers pump out chunks to peers with the playingback rate. So, except that the time for hop 1 is 1 time unit, the time for one hop is C/u_p. With the time for one hop, we can calculate the delay for a peer getting a chunk at hop i and maximum delay:

$$D(i) = 1 + \frac{C}{u_p} \cdot (i-1) \quad (3)$$

$$D_{max} = 1 + \frac{C}{u_p} \cdot \lceil log_{1+C} \frac{N}{u_s} \rceil \quad (4)$$

As the ISP-agnostic streaming pursues the best performance, it doesn't consider which ISP a peer is in. This makes peers randomly select other peers to build links. There is no optimization on the traffic. The probability that one peer connects to a peer from other peers is the same. And we assume that servers are deployed in ISP1. Servers pump out chunks only to peers in ISP1. Then, we can calculate the volumes of inter-ISP traffic:

For ISPm without deploying servers:

$$T_{mi} = [\frac{U_s}{N} + (R - \frac{U_s}{N}) \cdot \frac{N - N_m}{N-1}] \cdot N_m \quad (5)$$

$$T_{mo} = (R - \frac{U_s}{N}) \cdot \frac{N_m}{N-1} \cdot (N - N_m) \quad (6)$$

$$T_m = T_{mi} + T_{mo} \quad (7)$$

For ISP1 with servers:

$$T_{1i} = (R - \frac{U_s}{N}) \cdot \frac{N - N_1}{N-1} \cdot N_1 \quad (8)$$

$$T_{1o} = (R - \frac{U_s}{N}) \cdot \frac{N_1}{N-1} \cdot (N - N_1) \quad (9)$$

$$T_1 = T_{1i} + T_{1o} \quad (10)$$

Understanding the impact of traffic locality on performance:

We proceed to the analysis of locality-aware streaming protocols. Unlike the random peer selections and uncontrolled inter-ISP links in ISP-agnostic streaming protocols, locality-aware streaming protocols make controls on the inter-ISP links. And we apply the Inter-Peer model to quantify and analyze the controlled inter-ISP links. We first analyze the two-ISP case. The insight got from two-ISP case also holds true for multiple-ISP case.

The ISP without servers, ISP2, needs Inter-Peers to get chunks from ISP1. Let's see the case of one Inter-Peer that downloads one copy of streaming chunks from ISP1 and uploads them to ISP2, which is the maximum extent of traffic locality.

The Inter-Peer builds links with a peer in ISP1 having the chunk and with a peer in ISP2 that needs the chunk. Then, the chunk will be transmitted to a peer in ISP2 at hop d through the links built by the Inter-Peer. After that, the chunk will be disseminated to other peers in ISP2. We define a function $n_2(i)$ to express the dissemination of a chunk in ISP2:

$$n_2(i) = \begin{cases} 0, & \text{if } 1 \leq i \leq d-1 \\ (1+C)^{i-d}, & \text{if } d \leq i \end{cases}$$

The number of peers having a chunk in ISP2 after hop i is $N_2(i)$:

$$N_2(i) = min\{n_2(i), N_2\}$$

To let all the peers have the chunk, we need:

$$n_2(i) \geq N_2$$

So, the number of hops needed to disseminate a chunk is:

$$H_{max2} = d + \lceil log_{(1+C)} N_2 \rceil = 1 + \lceil log_{(1+C)} \frac{N_2}{\frac{1}{(1+C)^{d-1}}} \rceil$$

So, for peers in ISP2, the maximum delay is

$$D_{max2} = 1 + \frac{C}{u_p} \cdot \lceil log_{(1+C)} \frac{N_2}{\frac{1}{(1+C)^{d-1}}} \rceil$$

And for peers in ISP1, we use function $n_1(i)$ to express the chunk dissemination:

$$n_1(i) = \begin{cases} u_s(1+C)^{i-1}, & \text{if } 1 \leq i \leq d-1 \\ (u_s(1+C)^{d-1} - 1) \cdot (1+C)^{i-d}, & \text{if } d \leq i \end{cases} \quad (11)$$

The number of peers having a chunk in ISP1 after hop i is $N_1(i)$:

$$N_1(i) = min\{n_1(i), N_1\}$$

For all peers in ISP1 to get a chunk,

$$n_1(i) \geq N_1$$

And we can get the number of hops needed for all peers in ISP1 getting a chunk:

$$H_{max1} = 1 + \lceil log_{(1+C)} \frac{N_1}{u_s - \frac{1}{(1+C)^{d-1}}} \rceil$$

The maximum delay is:

$$D_{max1} = 1 + \frac{C}{u_p} \cdot \lceil log_{(1+C)} \frac{N_1}{u_s - \frac{1}{(1+C)^{d-1}}} \rceil$$

From the expressions for H_{max},H_{max1},H_{max2}, we can see that the sum of server capacities obtained by ISP1 and ISP2 equals to the total server capacity. The best strategy for ISP2 is that: when the peers in ISP1 get chunks pumped out from the servers, the Inter-Peer builds links with a peer in ISP1 with the chunk and with a peer in ISP2 that needs the chunk. Then, the chunk will be transmitted to a peer in ISP2 at hop $d = 2$ through the links built by the Inter-Peer. And, ISP2 gets $\frac{1}{1+C}$ relative server capacity.

If increasing the hop d at which that the Inter-Peer gets the chunk, the server capacity got from the Inter-Peer will decrease. There is a range of d, $2 \leq d \leq 1+\lceil log_{(1+C)} \frac{N_1}{u_s} \rceil + 1$. So, to use less inter-ISP traffic to obtain the needed server capacity, Inter-Peers should pull chunks from ISP1 at smaller hops.

From the analysis of one Inter-Peer case, we can see that besides increasing the total upload capacity of ISP2, the other function of Inter-Peers is to distribute the server capacity in different ISPs. This confirms that we can deploy servers in different ISPs to guarantee streaming performance with large extent of traffic locality. Or we need more controlled links by Inter-Peers to optimize the distribution of server capacity. We give the performances under n Inter-Peers.

For ISP1, the performance is:

$$H_{max1} = 1 + \lceil log_{(1+C)} \frac{N_1}{u_s - \sum_{i=1}^{n} \frac{1}{(1+C)^{d_i-1}}} \rceil$$

For ISP2, the performance is:

$$H_{max2} = 1 + \lceil log_{(1+C)} \frac{N_2}{\sum_{i=1}^{n} \frac{1}{(1+C)^{d_i-1}}} \rceil$$

The question we are interested is that: When do the locality-aware streaming protocols achieve the best performance? How much traffic are necessary to achieve it? It also equals to how many Inter-Peers we need.

We calculate how to optimize the distribution of server capacity to achieve the best performance. For the whole live streaming systems, the maximum chunk dissemination delay is $D_{max} = MaxD_{max1}, D_max2$. And, we use (u_{s1}, u_{s2}) to denote the distribution of server capacity.

$$u_{s1} + u_{s2} = u_s$$

$$D_{max1} = 1 + \frac{C}{u_p} \cdot \lceil log_{(1+C)} \frac{N_1}{u_{s1}} \rceil$$

$$D_{max2} = 1 + \frac{C}{u_p} \cdot \lceil log_{(1+C)} \frac{N_2}{u_{s2}} \rceil$$

So, as one of them increases, the other one decreases. To make D_{max} be the minimum,

$$D_{max1} = D_{max2}$$

$$1 + \frac{C}{u_p} \cdot \lceil log_{(1+C)} \frac{N_1}{u_{s1}} \rceil = 1 + \frac{C}{u_p} \cdot \lceil log_{(1+C)} \frac{N_2}{u_{s2}} \rceil$$

We get:

$$u_{s2} = \frac{u_s N_1}{N}$$

$$u_{s2} = \frac{u_s N_2}{N}$$

And the best performance is:

$$D_{max} = D_{max1} = D_{max2} = 1 + \frac{C}{u_p} \cdot \lceil log_{1+C} \frac{N}{u_s} \rceil$$

This result is the same with the snow-ball algorithm.

So, the necessary Inter-ISP traffic need to distribute the server capacity among ISPs to achieve the best system performance. Then, how can we use the least inter-ISP traffic to get the best performance? That also means how to use the fewest Inter-Peers to obtain the needed server capacity. From the analysis of one Inter-Peer case, we know that when the Inter-Peer is closer to servers, it obtains more server capacity. The closest distance of one Inter-Peer from servers is $d = 2$, and the server capacity it obtains is $\frac{1}{1+C}$.

Multiple-ISP case:

Now let's consider the M-ISP case. We assume servers are only deployed in ISP1. ISPi($2 \leq i \leq M$) needs to obtain server capacity through Inter-Peers. Then, the chunk dissemination delay of the whole system is $D_{max} = Max\{D_{max1}, ...D_{maxM}\}$. To make the D_{max} minimum, we need to let

$$D_{max1} = D_{max2} = ... = D_{maxM}$$

$$D_{maxi} = 1 + \frac{C}{u_p} \cdot \lceil log_{(1+C)} \frac{N_i}{u_{si}} \rceil$$

$$u_{si} = \frac{u_s N_i}{N}$$

B. Homogeneous & Unbalanced Case:

Different ISPs may provide users different types of network access. In this section, we analyze the homogeneous & unbalanced case, in which peers within the same ISP have the same bandwidth, and in different ISPs have different bandwidth.

Without loss of generality, we assume peers in ISP1 have larger bandwidth than peers in ISP2. Servers are deployed in ISP1 considering that servers usually serve peers with larger bandwidth.

Unlike the homogeneous & balanced case, in which there is no traffic locality without considering the IP information of peers as the streaming systems pursue the best performance, in homogeneous & unbalanced case, the heterogeneity of peers' bandwidth in different ISPs help streaming systems achieve a high level traffic locality as the streaming systems pursue good performance.

For good performance, the peers with larger bandwidth are put closer to servers and they transmit chunks to each other. After all the peers with larger bandwidth receive streaming contents from servers, peers with smaller bandwidth download chunks from them as free-riders.[] There are two situations:

1)When $N_2 \leq N_1 \cdot (u_{p1} - 1)$, the bandwidth of peers in ISP1 is enough to support peers in ISP2. All the peers in ISP2 are free-riders and download streaming chunks from ISP1. In this situation, the delay for all peers in ISP1 to get a chunk is:

$$D_1 = 1 + \frac{C}{u_{p1}} \cdot \lceil log_{(1+C)}(\frac{N_1}{u_s}) \rceil$$

After that, it will take 1 time unit for peers in ISP2 to download chunks from ISP1. So, the performance is:

$$D_{max} = 2 + \frac{C}{u_{p1}} \cdot \lceil log_{(1+C)}(\frac{N_1}{u_s}) \rceil$$

And meanwhile the streaming systems also obtain good traffic locality. Only peers in ISP2 download streaming contents from peers in ISP1. The inter-ISP traffic is:

$$T_{2i} = N_2 \cdot R$$

2)When $N_2 > N_1 \cdot (u_{p1} - 1)$, the bandwidth of peers in ISP1 isn't enough to support all peers in ISP2. Only a part of peers in ISP2 download streaming chunks from ISP1 and then serve other peers in ISP2. It takes delay D_1 for all peers in ISP1 to get the chunk. After that, it will take 1 time unit for $N_1 \cdot (u_{p1} - 1)$ peers in ISP2 to download chunks from ISP1, then these peers serve as servers to the other peers in ISP2. For the other peers in ISP2 to get the chunk, it takes time:

$$D_2 = \frac{C}{u_{p2}} \cdot \lceil log_{(1+C)} \frac{N_2}{N_1(u_{p1} - 1)}$$

So, the total delay is:

$$D_{max} = 2 + D_1 + D_2$$

Under this situation, peers in ISP2 can't get enough bandwidth from ISP1 to support the streaming, so they make use of their upload capacity to meet the bandwidth requirement. The inter-ISP traffic is:

$$T_{2i} = N_1 \cdot (u_{p1} - 1) \cdot R$$

From the above analysis, we see that the heterogeneity of peers' bandwidth in different ISPs contribute to traffic locality as the streaming systems try to achieve better performance, which let peers with larger bandwidth be closer to servers. However, the upload capacity of peers in ISP2 hasn't been fully utilized. It's a question whether we can achieve as good performance as the above results using less inter-ISP traffic. To achieve this, peers in ISP2 need to download chunks from ISP1 earlier to exploit their own upload capacities.

We assume that there are n Inter-Peers, so n copies of streaming chunks are sent to ISP2 at hop d. For peers in ISP1, it need i hops to let all peers get a chunk:

$$u_s(1 + C)^{i-1} - n(1 + C)^{i-d} \geq N_1 \quad (12)$$

$$i = 1 + \lceil log_{(1+C)} \frac{N_1}{u_s - \frac{n}{(1+C)^{d-1}}} \rceil \quad (13)$$

The delay for all peers in ISP1 getting the chunk is:

$$D_{max1} = 1 + \frac{C}{u_{p1}} \lceil log_{(1+C)} \frac{N_1}{u_s - \frac{n}{(1+C)^{d-1}}} \rceil$$

For peers in ISP2, after they get chunks from ISP1 after hop d, it need j more hops to let all peers in ISP2 get a chunk:

$$n(1 + C)^j \geq N_2 \quad (14)$$

$$j = \lceil log_{(1+C)\frac{N_2}{n}} \rceil \quad (15)$$

The delay for all peers in ISP2 getting the chunk is:

$$D_{max2} = 1 + \frac{C}{u_{p1}} \cdot d + \frac{C}{u_{p2}} \cdot \lceil log_{(1+C)\frac{N_2}{n}} \rceil$$

Not to make the performance worse, we should guarantee $D_{max1} \leq D_{max}; D_{max2} \leq D_{max}$.

V. Extensions to Heterogeneous case:

In the heterogeneous case, the peers are classified into different classes according to their upload capacities. We denote class i peer's upload capacity by U_i, and class i peer's relative capacity to streaming R is u_i. And $U_1 > U_2 > U_3 > ....$ In heterogeneous case, peers can be clustered based on their upload capacity. And the optimal topology of live streaming is to let peers with larger upload capacity be closer to servers.

One simple case is that there are two classes of peers in the streaming system: Super-peers and Free-riders. Suppose that there are N/u_1 super peers whose relative peer upload is u_1. The chunk could be disseminated among super peers by snow-ball algorithm. It takes time $\frac{1}{r} + \frac{\lceil \log_2 \frac{N}{u_1 u_s} \rceil}{r u_1}$ to let all super peers get the chunk. Then, the super peers could aggregately upload the chunk to $(1 - 1/u_1)N$ free-riders in time $(1/r - 1/ru_1)$. So, the total needed time for a chunk to be disseminated to all peers is $\frac{2}{r} + \frac{\lceil \log_2 \frac{N}{u_1 u_s} \rceil - 1}{r u_1}$. This is the delay for single chunk dissemination. Based on this single chunk dissemination, there exists a streaming algorithm to disseminate all chunks to all peers. And it has been proved that the delay bound for streaming is $\frac{2}{r} + \frac{\lceil \log_2 \frac{N}{u_1 u_s} \rceil}{r u_1}$.

For the case with multiple classes of peers, the chunk scheduling algorithm could be constructed iteratively based on the super-peer and free-rider case.

From the equations we could see that the delay is related to r, which is inverse proportional to chunk size. The equations show that the delay is proportional to chunk size. The reason is that when the chunk is small, more peers are able to contribute the bandwidth to upload chunks they have to others. In practice, we should select an appropriate size, the overheads for small size chunks are larger. Without loss of generality, we assume $r = 1$ in the following analysis.

In previous cases, we mainly focus on the homogeneous case where the peers in an ISP have the same uploading capacity. In real network environment, different peers have different types of network access, therefore, different uploading capacities. In this section, we study the impact of traffic locality on the performance under heterogeneous case.

To characterize the heterogeneity of peer upload capacity in real environment, we introduce three types of peers: super peers, ordinary peers and free-riders. Super peers represent the high bandwidth Ethernet users; ordinary peers represent the low bandwidth DSL users; free-riders represent those users that don't contribute their uploading capacity.

We denote the uploading capacity of super peers, ordinary peers, free-riders in ISPi by U_{isu}, U_{io}, U_{if} respectively.

We mainly divide the heterogeneous case into two cases: balanced case and unbalanced case. In balanced case, the average upload capacity is the same for all ISPs. In unbalanced case, the average upload capacity is different for different ISPs. In balanced case, the inter-ISP traffic is necessary for server capacity's distribution. In unbalanced case, the inter-ISP traffic is necessary for the uploading capacity's distribution and at the same time it can be used for the server capacity's distribution.

A. case 1: balanced case

B. case 2: unbalanced case

VI. Numerical results and insights

VII. Conclusion

The conclusion goes here.

References

[1] R. Kumar, Y. Liu, and K. Ross, "Stochastic Fluid Theory for P2P Streaming Systems," in *Proc. of IEEE INFOCOM 2007*, May 2007.

[2] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. Liu, and A. Silberschatz, "P4P: Provider Portal for Applications," in *Proc. of ACM SIGCOMM*, August 2008.

[3] *ISP Friend or Foe? Making P2P Live Streaming ISP-Aware*. Washington, DC, USA: IEEE Computer Society, 2009.

[4] *A Case Study of Traffic Locality in Internet P2P Live Streaming Systems*. Washington, DC, USA: IEEE Computer Society, 2009.