The Impact of Traffic Locality on Streaming Performance

***Abstract*—P2P streaming is now increasingly used for video broadcasting on Internet. Participating peers in P2P streaming contribute their upload capacities to serve one another and this greatly alleviates the requirement of server capacity. Much research attention has been paid on how to achieve a better streaming performance and the performance bound of live streaming. However, as P2P overlay network alleviates the burden of servers, it just move the streaming traffic burden to ISPs. The Inter-ISP traffic due to P2P applications has drawn the attention of ISPs. In this paper, we study what is the impact of traffic locality on the streaming performance. First we analyze how much inter-ISP traffic is brought by the snow-ball algorithm, which is one of the best algorithm for live streaming. Then, we use the Inter-Peer model to model the inter-ISP traffic. With the model, we analyze the function of inter-ISP traffic, and we show that except increasing the total upload bandwidth of an ISP, the other function of inter-ISP traffic is to obtain server capacities. And when the Inter-Peer is closer to servers, it obtains more server capacities.**

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 snow-ball 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

III. Problem Motivation and System Model

A. Problem Motivation

In previous work, researchers model the P2P live streaming system without considerations of traffic locality to explore the performance bounds of live streaming. As the increase of P2P traffic, ISPs are aware of the ISP-agnostic aspect of P2P applications and hope P2P applications become ISP-friendly. Then, the question is what is the impact of traffic locality on the performance of live streaming? The model of this paper is to answer this question.

B. System Model

In this section, we present our basic model under traffic locality, including the underlying assumptions and key notations. 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. 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 number of participating peers in ISP i is N_i. And the streaming rate is R.

Under traffic locality, the inter-ISP traffic is limited. In live streaming, every peer's downloading rate equals to playback rate R. So, we suppose that under traffic locality mechanism there are n_i Inter-Peers in ISP i, which only download chunks from other ISPs. Then, the flow-in traffic for ISP i is $n_i R$. The extent of locality is determined by the number of Inter-Peers n_i.

What's the impact of traffic locality on the performance of live streaming? We use the chunk dissemination delay as the metric for peers' performance. Live streaming is a delay sensitive application. The shorter the chunk dissemination delay is, the shorter the startup delay is.

Before the study of the impact of traffic locality on the performance, first, let's discuss the chunk dissemination delay without traffic locality. We discuss the problem under two cases: Homogeneous case and Heterogeneous case.

1) Homogeneous case: In the homogeneous 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. The optimal chunk dissemination delay could be achieved through snow-ball algorithm in the homogeneous case. And we assume that a peer uploads the chunk to children peers sequentially, which means that a peer uploads the chunk to another children peer only after finishing the present uploading. Then the optimal chunk dissemination delay through snow-ball algorithm is

$$D_{max}(N) = \frac{1}{r} + \frac{\lceil \log_2 \frac{N}{u_s} \rceil}{ru_p}$$

In one time unit, r chunks are played. If the relative server upload capacity and relative peer upload capacity is 1. The optimal delay is

$$D_{max}(N) = \frac{1}{r} + \frac{\lceil \log_2 N \rceil}{r}$$

2) 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 $\frac{1}{r} + \frac{\lceil \log_2 \frac{N}{u_1 u_s} \rceil}{ru_1}$ 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 $\frac{2}{r} + \frac{\lceil \log_2 \frac{N}{u_1 u_s} \rceil - 1}{ru_1}$. 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 $\frac{2}{r} + \frac{\lceil \log_2 \frac{N}{u_1 u_s} \rceil}{ru_1}$.

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.

IV. Study of impact of traffic locality on performance

In this section, we proceed to the impact of traffic locality on live streaming performance. We see in the previous section that under no traffic locality, the streaming system can apply snow-ball algorithm among peers with the homogeneous upload capacity or apply snow-ball algorithm iteratively among peers with different class of upload capacity to achieve the minimum delay bound. However, the snow-ball algorithm will bring on much traffic. Under traffic locality mechanism, to reduce the inter-ISP traffic, the snow-ball algorithm could only be applied among peers in one ISP. How will this inter-ISP traffic locality influence the chunk dissemination delay? To this end, we introduce the concept of Inter-Peers.

Inter-Peers and The model of traffic locality:

When there exists traffic locality, the inter-ISP traffic is limited. Here we introduce the concept of Inter-Peers to model the limited inter-ISP traffic. Under the traffic locality mechanism, some specific peers in an ISP are responsible to download chunks from other ISPs. Then the chunks can be disseminated to other peers in the same ISP. We call those specific peers Inter-Peers and assume that Inter-Peers only download chunks from other ISPs, then serve other peers in the same ISP. Because the downloading rate equals to the playback rate in live streaming, so, the inter-ISP traffic is determined by the number of Inter-Peers. We assume that there are n_{ij} peers in ISPi that only download chunks from peers in ISPj. The total number of Inter-Peers in ISPi is $n_i = \sum_{j=1}^{M} n_{ij}$, the flow-in traffic from other ISPs is $n_i R$. And the number of Inter-Peers in ISPj that try to download chunks from ISPi is n_{ji}. The total number of Inter-Peers in other ISPs that try to download chunks from ISPi is $n_i\prime = \sum_{j=1}^{M} n_{ji}$. Then the flow-out traffic from ISPi is $n_i\prime R$. Without loss of generality, the $n_i\prime$ Inter-Peers in other ISPs that download chunks from ISPi are equivalently free-riders in ISPi. And the n_i Inter-Peers in ISPi could be equivalent to increase n_i peers with upload capacity R as the sources of chunks for ISPi.

With n_i peers as the sources of chunks and $n_i\prime$ peers as free-riders, the total upload capacity of ISPi has increased by $(n_i - n_i\prime)R$ through inter-ISP traffic. One basic requirement for live streaming is that the total upload capacity should be greater than the total bandwidth consumption. So,

$$(n_i - n_i\prime)R + N_i\bar{U_{pi}} \geq N_i R$$

Through the above equation, it seems that the inter-ISP traffic that could support the total bandwidth consumption is enough. However, another requirement for live streaming is the real-time requirement. The chunk dissemination delay should

be short. The limit of inter-ISP traffic can make the chunk dissemination delay large for an ISP.

Understanding the impact of traffic locality on performance:

(1)Homogeneous case: We consider a streaming system with N peers. The peer relative upload capacity is $u_p > 1$, and server relative upload capacity is u_s. ISPi has N_i peers. We assume that servers are deployed in ISP1. Under no traffic locality, using snow-ball algorithm, the maximum chunk dissemination delay is $1 + \frac{\lceil \log_2 \frac{N}{u_s} \rceil}{u_p}$. Under traffic locality, the N peers are separated into two ISPs. The ISP without servers, ISP2, will need Inter-Peers to get chunks from outside. We assume that ISP2 has one Inter-Peer, which is the maximum extent of traffic locality. The best strategy for ISP2 is that: when the peers in ISP1 get chunks pumped out from the servers, they first push the chunks to the Inter-Peer in ISP2. Then, for Inter-Peer in ISP2, the delay of getting the chunk is $D_I = (1 + \frac{1}{u_p})$. Inter-Peer disseminates chunks among ISP2 after getting the chunks. So, for peers in ISP2, the maximum delay is

$$D_{max2} = D_I + \frac{\lceil \log_2(N_2) \rceil}{u_p} = 1 + \frac{\lceil \log_2 \frac{N_2}{0.5} \rceil}{u_p}$$

And for peers in ISP1, the server relative upload capacity is u_s, after D_I time, there will be $2u_s - 1$ peers in ISP1 that have the chunk. So we could divide the N_1 peers in ISP1 into $2u_s - 1$ groups, then, within each group, we could employ snow-ball approach to disseminate the chunk, the maximum delay for peers in ISP1 is:

$$D_{max1} = D_I + \frac{\lceil \log_2 \frac{N_1}{2u_s - 1} \rceil}{u_p} = 1 + \frac{\lceil \log_2 \frac{2N_1}{2u_s - 1} \rceil}{u_p} = 1 + \frac{\lceil \log_2 \frac{N_1}{u_s - 0.5} \rceil}{u_p}$$

From the above equations, we can see that under the best strategy for Inter-Peer of ISP2, the peers in ISP2 equivalently have 0.5 relative server capacity; and the relative server capacity in ISP1 equivalently decrease by 0.5.

Because servers can upload chunks to any peers in ISP1, so the best strategy needs the Inter-Peer to build connections with all peers in ISP1, when servers have pumped out the latest chunk to some peer, the Inter-Peer of ISP2 pull the chunk from the peer. In practical, Inter-Peer only has connections with a part of peers in ISP1, so, the delay for Inter-Peer getting a chunk D_I is among a range. The worst case is that the Inter-Peer in ISP2 get the chunk after all the peers in ISP1 get the chunk. Then, the delay for Inter-Peer getting the chunk will be $D_I = 1 + \frac{1 + \lceil \log_2 \frac{N_1}{u_s} \rceil}{u_p}$. For peers in ISP2, the maximum delay is:

$$\begin{aligned} D_{max2} &= D_I + \frac{\lceil \log_2 N_2 \rceil}{u_p} \\ &= 1 + \frac{1 + \lceil \log_2 \frac{N_1}{u_s} \rceil + \lceil \log_2 N_2 \rceil}{u_p} \\ &= 1 + \frac{t + \lceil \log_2 N_2 \rceil}{u_p} \\ &= 1 + \frac{\lceil \log_2 \frac{N_2}{\frac{1}{2^t}} \rceil}{u_p}; \\ t &= 1 + \lceil \log_2 \frac{N_1}{u_s} \rceil. \end{aligned}$$

In the worst case, the Inter-Peer has no effect on the dissemination of peers in ISP1. The chunk dissemination delay for peers in ISP1 is:

$$D_{max1} = 1 + \frac{\lceil \log_2 \frac{N_1}{u_s} \rceil}{u_p}$$

The delay for Inter-Peer in ISP2 getting a chunk is between the best case and the worst case $1 + \frac{1}{u_p} \leq D_I \leq 1 + \frac{\lceil log_2 \frac{N_1}{u_s} \rceil}{u_p}$. And the probability distribution is:

$$P_1(D_I = 1 + \frac{1}{u_p}) = 1 - \frac{C^{C_o} N_1 - u_s}{C^{C_o} N_1}$$

$$P_1(D_I = 1 + \frac{i}{u_p}) = (1 - \frac{C^{C_o} N_1 - 2(i-1)u_s}{C^{C_o} N_1}) \prod_{k=1}^{i-1} \frac{C^{C_o} N_1 - 2^{k-1} u_s}{C^{C_o} N_1}$$

$$P_1(D_I = 1 + \frac{\lceil log_2 \frac{N_1}{u_s} \rceil}{u_p}) = \prod_{k=1}^{\lceil log_2 \frac{N_1}{u_s} \rceil} \frac{C^{C_o} N_1 - 2^{k-1} u_s}{C^{C_o} N_1}$$

C_o is the number of partners of Inter-Peer in ISP2.

From the analysis of one Inter-Peer case, we could see that except increasing the total bandwidth of the ISP Inter-Peer is in, the other function of Inter-Peers is to serve as servers to make the chunk dissemination delay shorter, which is an important performance in live streaming systems.

The question we are interested is that: How much traffic are necessary to disseminate the chunk in the whole system best? It also equals to how many Inter-Peers we need in one ISP.

Let's also take the case of two ISPs to illustrate. First, we calculate how many server capacity are needed from ISP1 to ISP2, which is denoted by u_{s12}. For the whole live streaming system, the maximum chunk dissemination delay is $D_{max} = MaxD_{max1}, D_max2$.

$$D_{max1} = 1 + \frac{\lceil \log_2 \frac{N_1}{u_s - u_{s12}} \rceil}{u_p}$$

$$D_{max2} = 1 + \frac{\lceil \log_2 \frac{N_2}{u_{s12}} \rceil}{u_p}$$

To make D_{max} be the minimum,

$$\begin{aligned} D_{max1} &= D_{max2} \\ 1 + \frac{\lceil \log_2 \frac{N_1}{u_s - u_{s12}} \rceil}{u_p} &= 1 + \frac{\lceil \log_2 \frac{N_2}{u_{s12}} \rceil}{u_p} \\ u_{s12} &= \frac{u_s N_2}{N} \end{aligned}$$

And the best performance is:

$$D_{max} = D_{max1} = D_{max2} = 1 + \frac{\lceil log_2 \frac{N}{u_s} \rceil}{u_p}$$

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. We could see the chunk dissemination as a branching process. Here the distance of a peer from servers is the number of generation at which the peer gets the chunk. The closest distance of one Inter-Peer from servers is $d = 2$, and the server capacity it obtains is $\frac{1}{2^{d-1}}$.

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 $D_{max} = MaxD_{max1}, ...D_{maxM}$. To make the D_{max} minimum, we need to let

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

$$D_{max1} = 1 + \frac{\lceil \log_2 \frac{N_1}{u_s - \sum_{i=2}^{M} u_{s1i}} \rceil}{u_p}$$

$$D_{maxi} = 1 + \frac{\lceil \log_2 \frac{N_i}{u_{s1i}} \rceil}{u_p}$$

$$u_{s1i} = \frac{u_s N_i}{N}$$

(2)Heterogeneous case:

V. SIMULATIONS

VI. CONCLUSION

The conclusion goes here.

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

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

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

[3] D. R. Choffnes and F. E. Bustamante, "Taming the Torrent: A Practical Approach to Reducing Cross-ISP Traffic in Peer-to-Peer Systems," in *Proc. of ACM SIGCOMM*, August 2008.

[4] S. Le Blond, A. Legout, W. Dabbous, and I. Mediterranee, "Pushing BitTorrent Locality to the Limit," *Arxiv preprint arXiv:0812.0581*, vol. 0, p. 0, 2008.

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

[1] [2] [3] [4] [5]