Weekly Report

Jian Zhao

June 3, 2010

Analysis of Locality of P2P live streaming

The streaming capacity of P2P streaming system with locality limit:
Before we discuss the impact of traffic locality on the performance of P2P live streaming systems, let's see the optimal performance of a live streaming system without locality.

We intend to model the live streaming system as a discrete time system. Due to the reason that the size of chunks can be changed, we suppose a time unit is a fixed length of time.

The maximum streaming rate of a P2P streaming system is:

$$R_{max} = min U_s, \frac{1}{N}(U_s + \sum_{i=1}^{N} U_i)$$

Because the servers' upload capacity is usually powerful enough. So,

$$R_{max} = \frac{1}{N}(U_s + \sum_{i=1}^{N} U_i) = \frac{U_s}{N} + U_a$$

Then, let's see the minimum delay of the streaming. The P2P streaming is to delivery a chunk to every peer. We first see the minimum delay for one chunk's transmission to every peer. We divide this question into two parts: 1) Homogeneous case; 2) Heterogenous case; 1)Homogeneous case: All the peers' upload capacity are the same, denoted by U_p. The maximum streaming rate that could be sustained is:

$$R_{max} = \frac{U_s}{N} + U_p$$

If N tends to infinity, $R_{max} = U_p$. If we assume the streaming playback rate is R, then the average upload capacity of the system is larger than or equal to the playback rate. We set the size of the chunks so that in one time unit, r chunks are played.

We first assume that the server and the peer's bandwidth is 1, which means the server and the peer bandwidth equal to the playback rate. So, in one time unit, the server and peer could upload r chunks. Using snow-ball chunk dissemination, the maximum delay for all peers to receive a chunk is:

$$D_{pmax} = 1 + \lceil \log_{(r+1)} N \rceil$$

The average delay is:

$$\bar{D_p} = \frac{1}{N}\{1 + \sum_{i=2}^{D_{max}-1} ir(r+1)^{i-2} + D_{max}[N - (r+1)^{D_{max}-2}]\}$$

When a peer uploads the chunk simultaneously to r children peers, and when a peer uploads the chunk to children peers sequentially, the maximum delay for all peers to receive a chunk is:

$$D_{smax} = \frac{1 + \lceil \log_2 N \rceil}{r}$$

The average delay is:

$$\bar{D_s} = \frac{1}{r \cdot N}[1 + \sum_{i=2}^{K^*-1} i2^{i-2} + K^*(N - 2^{K^*-2})]$$

The effect of increasing Server bandwidth:

What's the impact of increasing server bandwidth from 1 to u_s on the streaming performance? The maximum streaming rate is $R_{max} = \frac{U_s}{N} + U_p$. The maximum streaming rate just increases by $\frac{1}{N}$ of the increasing part of the server bandwidth. On the other hand, what is the improvement on the delay? Let z(k) be the number of peers (including the server) with the chunk at the beginning of time unit k. Then,

$$z(0) = 1; z(1) = 1 + u_s; z(k) = r \cdot [z(k-1) - 1] + z(k-1)$$

$$z(k) - 1 = (r+1)[z(k-1) - 1]$$

We need $z(k) \geq N$, so $D_{pmax} = k = \lceil log_{r+1} \frac{N}{u_s} \rceil + 1$. This delay improvement could be reached. We can divide N peers into u_s groups, the server simultaneously upload the chunk to one peer in each group within one time unit. Then, in each group, the peers could use snow-ball algorithm to distribute the chunk, the delay for the chunk dissemination will be $1 + \lceil log_{r+1} \frac{N}{u_s} \rceil$.

The average delay under this situation is:

$$\bar{D_p} = \frac{1}{N}\{u_s \cdot 1 + \sum_{i=2}^{D_{pmax}-1} ir(r+1)^{i-2}u_s + D_{pmax}[N - u_s(r+1)^{D_{pmax}-2}]\}$$

If the peers take sequential uploading, transmitting the chunk to another peer only after finishing the current transmitting, the improvement on the delay is calculated as follows:

$$z(k) = 2^r[z(k-1) - 1]$$

$$z(1) = 2^{(r-1)}u_s + 1$$

$$D_{smax} = k = \frac{1 + \lceil log_2 \frac{N}{u_s} \rceil}{r}$$

The average delay under this situation is:

$$\bar{D_s} = \frac{1}{r \cdot N}\{u_s + \sum_{i=2}^{D_{smax}-1} i2^{i-2}u_s + D_{smax}[N - 2^{D_{smax}-2}u_s]\}$$

The effect of increasing peer bandwidth If we increase the peer bandwidth from 1 to u_p, then, every time unit one peer could upload u_p chunks.

$$D_{pmax} = 1 + \lceil \log_(ru_p + 1)N \rceil$$

$$D_{smax} = \frac{1 + \lceil \log_2 N \rceil}{ru_p}$$

The relationship between Streaming Rate and Delay. One peer's buffer size B should be no less than the maximum delay D. Because the sequential uploading is shorter in delay, we take the sequential uploading.So,

$$D_{smax} < B$$

the impact of locality on the delay performance

What is the impact of locality on the delay performance?

In the snow-ball algorithm, the traffic?

If the peers are distributed in different ISPs, how much inter-ISP traffic will the snow-ball algorithm bring? We suppose there are N peers in total, and in ISP1 there are N_1 peers. The servers' capacity could support u_s peers. So, the servers could upload u_s copies of contents. Then, the N peers need to upload $N-u_s$ copies of contents. In average, every peer uploads $\frac{N-u_s}{N}$ copies of contents. N_1 peers in ISP1 upload $\frac{N_1(N-u_s)}{N}$ copies of contents and among them, $\frac{N-N_1}{N}$ parts flow out of ISP1. So, under snow-ball algorithm, the inter-ISP traffic that flow into is $\frac{N_1(N-N_1)(N-u_s)}{N^2}$.

under traffic limit, what's the delay?

Snow-ball algorithm brings more than one copy of contents into one ISP from other ISPs. This is the cause of the sharply increased inter-ISP traffic in the internet. Here we intend to reduce the inter-ISP traffic. One extreme case is that peers in one ISP just fetch one copy of contents from other ISPs. Without loss of generality, we assume the servers are deployed in ISP1, and peers in ISP2 just fetch one copy of streaming contents from ISP1 and then distribute it within ISP2. The inter-ISP traffic is reduced to the minimum. But what about the performance? let us calculate the chunk dissemination delay for peers in ISP2. For peers in ISP2 to fetch one chunk from ISP1, the average delay that the chunk could be got is $D_{1\bar{s}max}$.

2) Heterogenous case:

Analysis of decentralized algorithm with locality mechanism

In pull-based live streaming system, one peer exchanges its bitmap with its partners periodically. After getting its partners' bitmap, it will be able to send requests for the chunks that its partners have but it doesn't have. The priority for requesting different chunks are determined by chunk scheduling strategy, which may be different for different live streaming systems. When a peer receives the requests for chunk transmissions from its partners, it will assign its upload capacity among its partners according to some peer selection strategy.

One simple way to improve the locality of P2P live streaming is biased peer selection. In biased peer selection, one peer only has a fixed number of partners in other ISPs. And most of the partners are from the same ISP.

We first analyze the inter-ISP traffic of P2P live streaming under random peer selection strategy and latest useful chunk selection strategy. Then, we consider the inter-ISP traffic under some locality mechanism: biased peer selection. Compare the performance and inter-ISP traffic without locality mechanism and with locality mechanism. We could get the insight of the trade off between the performance and inter-ISP traffic.

We analyze three cases: Homogeneous case, Heterogeneous case, peer churn case.

Notation Table

Symbol	Meaning
P(i)	the probability that a peer's buffer unit i is filled
$P_m(i)$	the probability that the buffer unit i of one peer in ISPm is filled
Q(i)	the probability that a peer gets the chunk from other peers to fill up buffer unit i+1
$Q_m(i)$	the probability that a peer in ISPm gets the chunk to fill up the buffer unit i+1 from other peers.
$q(i)$	the probability that a peer gets the chunk to fill up buffer unit i+1 from one connection
$q_k^{mn}(i)$	the probability that a peer in ISPm gets the chunk to fill up buffer unit i+1 from a peer with upload capacity U^k in ISPn.
s(i)	the probability that a peer's partner selects chunk i to upload to the peer.
$s_k^{mn}(i)$	the probability that a partner in ISPn selects chunk i to upload to the peer in ISPm.
C	the number of partners of a peer.
C_o	the number of partners of a peer that aren't in the same ISP.
T_d	the average download traffic of a peer.
T_i	the average traffic that flow into an ISP from other ISPs.
T_o	the average traffic that flow out of an ISP.
T_b	the total traffic that flow into and out of an ISP.
N	the number of peers in the systems
p^{ji}	the probability of class i peers in ISPj.
p^i	the distribution probability of class i peers in the streaming system.

Homogeneous case

In homogeneous case, the environment of P2P live streaming is: N_1 peers in ISP_1,N_2 peers in other ISPs, which are outside ISP_1; the upload capacity is assumed homogeneous, suppose the upload capacity u_p is equal to the playback rate r. The buffer has n units. The probability that buffer unit i is filled by a chunk is $P_1(i)$ for peers in ISP_1 and $P_2(i)$ for peers in other ISPs. One chunk will be played in one time slot. The chunk in buffer unit n is playing. And due to the upload capacity, no more than one chunk will be uploaded by a peer in one time slot. The server's capacity in ISP_1 is u^{s1}. Each peer has n partners chosen from their peer list.

Situation with no locality mechanism

First, we analyze the performance and inter-ISP traffic under no locality mechanism situation.

The performance is measured by the probability that buffer units are filled,$P(i)$. For one buffer unit i, the probability that it is filled is $P(i+1) = P(i) + Q(i)$. $Q(i)$ is the probability that one peer download chunk i+1 to fill buffer unit i+1 in one time unit.

Let's consider one connection between one peer and one of its partners. The probability that the peer gets chunk i+1 from a partner is:

$$q(i) = Pr[H_p(i)] \times Pr[W(i)] \times Pr[S(i)|H_p(i), W(i)]$$

$$q(i) = P(i)[1 - P(i)]s(i)$$

$$s(i) = s(1) + \frac{1}{C}P(1) - \frac{1}{C}P(i)$$

We assume one peer has C partners. Then,

$$Q(i) = 1 - [1 - q(i)]^C$$

So,

$$P(i+1) = P(i) + 1 - [1 - q(i)]^C$$

Based on this, let's consider the inter-ISP traffic under no locality mechanism. We assume N_1 peers in ISP_1 and the upload capacity of servers in ISP1 could support N^{s1} peers, N_2 peers in other ISPs and the upload capacity of servers in other ISPs could support N^{s2} peers. In one time unit, the download traffic of one peer is

$$T_d = \sum_{i=1}^{i=n-1} Q(i) + \frac{N^{s1} + N^{s2}}{N_1 + N_2}$$

Then, the inter-ISP traffic from other ISPs to ISP1 will be

$$T_i = \frac{N_2 N_1}{N_1 + N_2} \times [\sum_{i=1}^{i=n-1} Q(i)] + \frac{N_1 N^{s2}}{N_1 + N_2}$$

And the inter-ISP traffic from ISP1 to other ISPs will be

$$T_o = \frac{N_2 N_1}{N_1 + N_2} \times [\sum_{i=1}^{i=n-1} Q(i)] + \frac{N_2 N^{s1}}{N_1 + N_2}$$

So the total inter-ISP traffic will be

$$T_b = T_i + T_o$$

Situation with locality mechanism

After the situation with no locality mechanism, let's see the situation with locality mechanism. In the locality mechanism, we let one peer have a fixed number of outer-the same ISP partners,C_o.

We assume the probability that the buffer unit i is filled in ISP1 is $P_1(i)$, and $P_2(i)$ for ISP2. Then,

$$P_1(i+1) = P_1(i) + Q_1(i)$$
$$P_2(i+1) = P_2(i) + Q_2(i)$$

$$Q_1(i) = 1 - [1 - q_{11}(i)]^{C-C_o}[1 - q_{12}(i)]^{C_o}$$
$$Q_2(i) = 1 - [1 - q_{22}(i)]^{C-C_o}[1 - q_{21}(i)]^{C_o}$$

$$q_{11}(i) = [1 - P_1(i)]P_1(i)s_{11}(i)$$
$$q_{12}(i) = [1 - P_1(i)]P_2(i)s_{12}(i)$$
$$q_{21}(i) = [1 - P_2(i)]P_1(i)s_{21}(i)$$
$$q_{22}(i) = [1 - P_2(i)]P_2(i)s_{22}(i)$$

$$s_{11}(i) = \frac{1}{C}(1 - \frac{N^{s1}}{N_1}) \prod_{j=1}^{i-1} [P_1(j) + (1 - P_1(j))^2]$$

$$s_{12}(i) = \frac{1}{C}(1 - \frac{N^{s2}}{N_2}) \prod_{j=1}^{i-1} [P_1(j) + (1 - P_1(j))(1 - P_2(j))]$$

$$s_{21}(i) = \frac{1}{C}(1 - \frac{N^{s1}}{N_1}) \prod_{j=1}^{i-1}[P_2(j) + (1 - P_2(j))(1 - P_1(j))]$$

$$s_{22}(i) = \frac{1}{C}(1 - \frac{N^{s2}}{N_2}) \prod_{j=1}^{i-1}[P_2(j) + (1 - P_2(j))^2]$$

Then, let's consider the traffic under this situation:

$$T_{d1} = \sum_{i=1}^{n-1} Q_1(i) + \frac{N^{s1} + N^{s2}}{N_1 + N_2}$$

$$T_{i1} = N_1 \sum_{i=1}^{n-1} [1 - q_{11}(i)]^{(C-C_o)} - [1 - q_{11}(i)]^{(C-C_o)}[1 - q_{12}(i)]^{C_o}$$

$$T_{o1} = N_2 \sum_{i=1}^{n-1} [1 - q_{22}(i)]^{C-C_o} - [1 - q_{22}(i)]^{C-C_o}[1 - q_{21}(i)]^{C_o}$$

$$T_b = T_{i1} + T_{o1}$$

Heterogeneous Case

Situation with no locality mechanism

In the heterogeneous case(in terms of peers' bandwidth), the peers' bandwidth are different. Assume there are h classes of peers. The upload capacity of each class is U^i, and the download capacity of each class is D^i, $U^1 < U^2 < U^3 < ... < U^r < ... < U^h$, $D^1 < D^2 < D^3 < ... < D^r < ... < D^h$. Let p^i denote the percentage of class i peers in the streaming system. We model the live streaming system as a discrete time system. At the beginning of one time slot, peers exchange bitmap with partners. Assume in one time slot class i peer(upload capacity U^i) could only transmit i chunks. Denote the playback rate of the streaming video by r chunks per time slot. The downloading rate of peers is around r chunks per time slot. And the upload capacity are the bottlenecks. So, we assume the uploading rates of peers are $u^i = U^i$ for class i peers.

We assume that every peer exchanges its bitmap with C partners every time slot. Then, the peer requests for the missing chunks in its buffer from its partners that have the chunks. After receiving the requests, the partners schedule the sending of chunks. One partner may receive requests for chunk transmissions from many peers. With the limit of upload capacity, we assume that in one time slot, the partner randomly selects one peer to transmit chunks. So, the probability for one partner sending chunks to a peer according to the requests will be $1/C$.

The upload capacity of the server is u^s, the new content rate pumped into the system by the server equals to the playback rate r, and the upload capacity of the server could support N^s peers simultaneously.

Now let's derive the probability that the buffer is filled in the heterogeneous case.

First, consider one connection between a peer with one of its partners. The probability that buffer unit i+r of the peer is filled is

$$P(i + 1) = P(i) + Q(i)$$

Here P(i) is the probability that buffer unit i of the peer is filled. Q(i) is the probability that buffer unit i+r get the chunk to fill it up from other peers. The next step is to derive Q(i), using P(i) to express Q(i), then, we could get a recursive equation for P(i).

When one peer connects to a partner of class k, the probability that the peer could get chunk i is:

$$q_k(i) = [1 - P(i)]P(i)s_k(i)$$

$s_k(i)$ is the probability that the partner of class k transmits chunk i to the peer. And we could derive that:

$$s_k(i) = \frac{k}{C}\prod_{j=1}^{i-1}[P(j) + [1 - P(j)]^2]$$

So, for one connection, the probability that one peer gets chunk i is q(i), $q(i) = (1-\frac{N^s}{N})[p^1q_1(i) + p^2q_2(i) + p^3q_3(i) + ... + p^hq_h(i)]$. Here, $1-\frac{N^s}{N}$ means the probability that the connection isn't between the peer with the server. When the peer connects to the server, the server will send the new contents to the peer.

$$Q(i) = 1 - [1 - q(i)]^C$$

Then, let's consider the traffic under this situation: The download traffic for one peer is:

$$T_d = \sum_{i=1}^{i=n-1} Q(i) + \frac{N^s}{N}$$

We assume ISP1 has N_1 peers. And the probability distribution of N_1 peers' bandwidth is $p_{1i}, 1 \leq i \leq h$. Then, the total upload capacity in ISP1 is $N_1\sum_{i=1}^{h} p_{1i}U^i$. The total upload capacity in the streaming system is $N\sum_{i=1}^{h} p_iU^i$.

The inter-ISP traffic from other ISPs to ISP1 is:

$$T_i = N_1[1 - \frac{N_1\sum_{i=1}^{h} p_{1i}U^i}{N\sum_{i=1}^{h} p_iU^i}]\sum_{i=1}^{i=nr-1} Q(i) + \frac{N - N^{s1}}{N}$$

The inter-ISP traffic from ISP1 to other ISPs is:

$$T_o = (N - N_1)\frac{N_1\sum_{i=1}^{h} p_{1i}U^i}{N\sum_{i=1}^{h} p_iU^i}\sum_{i=1}^{i=nr-1} Q(i) + \frac{N^{s1}}{N}$$

$$T_b = T_i + T_o$$

Situation with locality mechanism

In the situation with locality mechanism, the distributed locality mechanism is that one peer has a fixed number of partners in different ISPs, fewer than the partners in the same ISP. We denote the number of inter-ISP partners by C_o. And due to the locality, the performance of peers in different ISPs may be different, so we assume the probability that the buffer unit i is filled in ISP1 is $P_1(i)$, and $P_2(i)$ for ISP2. Then,

$$P_1(i + 1) = P_1(i) + Q_1(i)$$

$$P_2(i + 1) = P_2(i) + Q_2(i)$$

$$Q_1(i) = 1 - [1 - q_{11}]^{C-C_o}[1 - q_{12}]^{C_o}$$

$$Q_2(i) = 1 - [1 - q_{22}]^{C-C_o}[1 - q_{21}]^{C_o}$$

$$q_{11}(i) = (1 - \frac{N^s}{N})[p^{11}q_1^{11}(i) + p^{12}q_2^{11}(i) + p^{13}q_3^{11}(i) + ... + p^{1h}q_h^{11}(i)]$$

$$q_{12}(i) = (1 - \frac{N^s}{N})[p^{21}q_1^{12}(i) + p^{22}q_2^{12}(i) + p^{23}q_3^{12}(i) + ... + p^{2h}q_h^{12}(i)]$$

$$q_{22}(i) = (1 - \frac{N^s}{N})[p^{21}q_1^{22}(i) + p^{22}q_2^{22}(i) + p^{23}q_3^{22}(i) + ... + p^{2h}q_h^{22}(i)]$$

$$q_{21}(i) = (1 - \frac{N^s}{N})[p^{11}q_1^{21}(i) + p^{12}q_2^{21}(i) + p^{13}q_3^{21}(i) + ... + p^{1h}q_h^{21}(i)]$$

$$q_k^{11}(i) = [1 - P_1(i)]P_1(i)\frac{k}{C}\prod_{j=1}^{i-1}[P_1(j) + [1 - P_1(j)]^2]$$

$$q_k^{12}(i) = [1 - P_1(i)]P_2(i)\frac{k}{C}\prod_{j=1}^{i-1}[P_1(j) + [1 - P_1(j)][1 - P_2(j)]]$$

$$q_k^{22}(i) = [1 - P_2(i)]P_2(i)\frac{k}{C}\prod_{j=1}^{i-1}[P_2(j) + [1 - P_2(j)]^2]$$

$$q_k^{21}(i) = [1 - P_2(i)]P_1(i)\frac{k}{C}\prod_{j=1}^{i-1}[P_2(j) + [1 - P_2(j)][1 - P_1(j)]]$$

The traffic:

$$T_{d1} = \sum_{i=1}^{n-1} Q_1(i) + \frac{N^{s1} + N^{s2}}{N_1 + N_2}$$

$$T_{i1} = N_1 \sum_{i=1}^{n-1} [1 - q_{11}(i)]^{(C-C_o)} - [1 - q_{11}(i)]^{(C-C_o)}[1 - q_{12}(i)]^{C_o}$$

$$T_{o1} = N_2 \sum_{i=1}^{n-1} [1 - q_{22}(i)]^{C-C_o} - [1 - q_{22}(i)]^{C-C_o}[1 - q_{21}(i)]^{C_o}$$

$$T_b = T_{i1} + T_{o1}$$

peer churn case

In mesh-based P2P live streaming systems, peer dynamics will make the number of one peer's partners changing. So, under dynamic situation, the number of one peer's partners is not fixed. We could assume the number of partners is between C_{min} and C_{max}. Then, we need to derive the probability that one peer has $c, C_{min} \leq c \leq C_{max}$ partners.

The service is distributed in every peer. The service capacity of a peer is related both to the upload capacity and the content availability. What do I want to analyze using the queuing theory? How will the peer churn influence the chunk dissemination delay? What is the service and what are the customers?