# Stochastic Model for ISP-aware VoD Streaming

Jian Zhao, Chuan Wu
Department of Computer Science
The University of Hong Kong, Hong Kong
*jzhao, cwu@cs.hku.hk*

Xiaojun Lin
School of Electrical and Computer Engineering
Purdue University, West Lafayette
*linx@purdue.edu*

***Abstract*—**

## I. Introduction

People start to enjoy watching online videos due to the proliferation of high-speed broadband services. The video traffic in the Internet and the server workloads increase largely. The content distributors apply peer-to-peer technology in Video-on-Demand systems, (*e.g.*, PPLive [1], UUSee [2]), to alleviate the heavy workload of servers in data centers. Distributed peers' storage and upload bandwidth resources are exploited in P2P technology, which increases the inter-ISP traffic inevitably. The content distributors strive to achieve high quality and smooth video and low server workloads without considering the ISP awareness. The ISP-agnostic P2P connections bring about large volume of unnecessary inter-ISP traffic, which increases the cost of ISPs. This makes ISPs start to proactively detect and throttle P2P data packets, which definitely affects the service quality.

To solve the tussle between ISPs and content distributors, ISP-aware P2P applications are proposed. P4P [3] achieves ISP-friendly traffic control based on an architecture providing interfaces for networks to communicate with P2P applications. Huang *et al.* [4] design distributed peer selection algorithms that can effectively achieve any desired performance and locality tradeoff through multi-objective optimization. Fabio Picconi *et al.* [5] proposes a two-level overlay and a dynamic unchoke algorithm to reduce unnecessary inter-ISP traffic in P2P live streaming applications. Wang *et al.* [6] design an ISP-friendly rate allocation algorithm for peer-assisted VoD.

To become ISP-aware and avoid the blocking of ISPs, peers control the cross-ISP connections to reduce the unnecessary cross-ISP traffic. How will this control affect the system performance? The impact is not well understood. There is still lack of theoretical study on the impact of controlled peer selection on P2P system performance. This paper focuses on peer-to-peer Video-on-Demand systems. We characterize ISP-awareness through the number of peers' inter-ISP connections. The chunk request rates in each ISP are derived based on the number of peers' inter-ISP connections. With chunk request rates in each ISP, we apply the loss network model to analyze the chunk loss rates, which are the solutions of an optimization problem under a specific peer cache distribution. We prove that the solutions of the optimization problem can be mapped into the corresponding maximum bipartite flow. A modified push-rebel algorithm is presented to solve it. Through the algorithm, we get the optimal peer cache placement strategy and the corresponding analytical results for chunk loss rates. We also perform simulations to validate our theoretical study.

The remainder of the paper is organized as follows. We present our system model and notations in Sec. II and apply the stochastic loss network analysis in Sec. III. We map the solutions of the optimization problem obtained from the loss network analysis into the corresponding maximum bipartite flow and design an algorithm to solve it in Sec. **??**. We state the optimal peer cache placement strategy and the corresponding analytical chunk loss rates in Sec. IV. We perform performance evaluation in Sec. VI, and conclude the paper in Sec. VII.

## II. Model and Notation

We first introduce the P2P VoD system model.

We consider a VoD system involving $M$ ISPs. ISP $m$ has totally $N_m$ participating peers in the VoD system. The average peer upload bandwidth in ISP $m$ is $U_m$. The VoD system supplies multiple video channels. As a peer can watch any chunks in any video at a time, we consider a collection of $J = |\mathcal{C}|$ chunks, $\mathcal{C} = \{c_1, c_2, ..., c_J\}$, regardless of which video they belong to, instead of the channels peers are watching. The playback time for one chunk is one unit time. A peer can cache and serve chunks from different videos. Every peer has a cache size $B$. $s_i = \{c_1^{(i)}, c_2^{(i)}, ..., c_B^{(i)}\}$ denotes the cache state $i$. Let $\Theta$ be the set of all different cache states of peers, $W = |\Theta|$. The servers have cached all chunks.

### A. Peers' Cache Distribution

In the VoD system, a peer contributes a storage of size $B$ to cache chunks. The state of a peer's cache is $s_i = \{c_1^{(i)}, c_2^{(i)}, ..., c_B^{(i)}\}, 1 \leq i \leq W$. $N_m^{(i)}$ denotes the number of peers with cache state $s_i$ in ISP $m$. The stationary distribution of cache states under a specific cache placement strategy is $\gamma_i$, for state $s_i$. When the peer number in ISP $m$, $N_m$, is large enough, $N_m^{(i)} = \gamma_i \cdot N_m$. The proportion of peers caching chunk $c_j$ is $\rho_j = \sum_{i:c_j \in s_i} \gamma_i$. The number of peers caching chunk $c_j$ is $N_m \cdot \rho_j$.

### B. Chunk Request

In the VoD system, different users may watch different channels or different parts of videos. Hence, they may be downloading different video chunks. When peers replay the watched and cached parts, they will not need to download new chunks. Let $\phi$ denote the probability that a peer replays watched video and do not need to download new chunks. We define chunk $j$'s popularity when peers download new chunks, the probability that peers in the VoD system are downloading chunk $j$. $(\pi_1, \pi_2, \pi_3, \ldots, \pi_J)$ denote the chunk popularity. A request for a chunk is generated when a peer selects to download the chunk. As peers' playback rate is 1 chunk per unit time, the request rate is at least 1 request per unit time to catch up with the playback. We assume a peer's request rate is 1 request per unit time. The requests for chunks generated by peers in ISP $m$ are the superposition of $N_m$ peers' requests for chunks. As $N_m$ is large, one peer's number of requests for chunks is a general renewal process with relative small intensity. According to Palm-Khintchine theorem [7], the requests for chunks generated by peers in ISP $m$ can be modeled as a Poisson Process, with request rate $\lambda_m = N_m$. Given that a peer who has cached chunk $j$ requests for chunk $j$, the request can be served by its own cache, no downloading is necessary, with chunk $j$'s popularity, the request rate for chunk $j$ is $r_{m,j} = \lambda_m \cdot \pi_j(1-\phi)$. The total request rate generated by peers in ISP $m$ that needs downloading chunks is $r_m = \sum_{j=1}^{J} r_{m,j} = \lambda_m \sum_{j=1}^{J} \pi_j \cdot (1-\phi) = (1-\phi)\lambda_m$.

Let $a_{ml}$ denotes the proportion of chunk requests routed from ISP $m$ to ISP $l$. The requests for chunk $j$ routed into ISP $m$ is

$$\nu_{m,j} = \sum_{l=1}^{M} a_{lm} \cdot r_{l,j}$$

The total chunk requests routed into ISP $m$ is

$$\nu_m = \sum_{l=1}^{M} a_{lm} \cdot r_l.$$

### C. Loss Probabilities

The VoD streaming is a delay-sensitive application. The requests that can not be served by peers' resource are redirected to servers. Let $L_{m,j}$ be the loss probability of request for chunk $j$ in ISP $m$, *i.e.*, the steady state probability that a request for chunk $j$ routed to ISP $m$ is dropped and redirected to servers.

The average loss probability of requests in ISP $m$ is

$$L_m = \frac{\sum_{j=1}^{J} L_{m,j}\nu_{m,j}}{\nu_m}$$

The total loss probability in the VoD system is

$$L = \frac{\sum_{m=1}^{M} L_m \cdot \nu_m}{\sum_{m=1}^{M} \nu_m}$$

### D. Cross-ISP traffic

When serving a chunk request from other ISPs, cross-ISP traffic will be generated. The cross-ISP traffic from other ISPs to ISP $m$ is

$$T_m^i = \sum_{l=1, l\neq m}^{M} a_{ml} \cdot r_m \cdot (1 - L_l).$$

By summing up the cross-ISP traffic into all ISPs, we get the total cross-ISP traffic:

$$T = \sum_{m=1}^{M} T_m^i$$

We summarize important notations in Table I for ease of reference.

TABLE I.
IMPORTANT NOTATIONS

| | |
|---|---|
| $N$ | total number of peers in the system. |
| $M$ | number of ISPs. |
| $N_m$ | number of peers in ISP $m$. |
| $N_m^{(i)}$ | number of peers in ISP $m$ with cache state $s_i$. |
| $U_m$ | average peer upload bandwidth in ISP $m$. |
| $B$ | the cache size of a peer. |
| $\mathcal{C}$ | the set of all chunks shared in VoD system. |
| $J$ | the number of chunks shared in VoD. |
| $\Theta$ | the set of all possible cache states of peers. |
| $W$ | the number of different cache states. |
| $\rho_j$ | the proportion of peers that have cached chunk $j$. |
| $r_{m,j}$ | the request rate for chunk $j$ generated by peers in ISP $m$. |
| $\nu_{m,j}$ | the request rate for chunk $j$ routed into ISP $m$. |
| $a_{lm}$ | the fraction of requests routed from ISP $l$ to ISP $m$. |
| $\phi$ | the probability that a peer's chunk requests are not served by itself cache. |
| $L_{m,j}$ | the loss rate for chunk $j$ in ISP $m$. |
| $T_m^i$ | the cross-ISP traffic flowing into ISP $m$. |
| $T$ | the total cross-ISP traffic in VoD system. |

## III. A Model Framework for P2P VoD Systems

### A. Loss Network Model

The acceptance and rejection of chunk requests in the P2P VoD system can be modeled as a loss network, which suits the characteristics of zero waiting time for requests in VoD applications [8] [9]. Compared with the basic model of a loss network with terminology based on routes and links, the requests for different chunks correspond to the calls on different routers, the peers with different cache states correspond to the different links. Peers' upload bandwidth correspond to the circuits of a link. The requests for a chunk can link to peers caching the chunk for service. The service time is one unit time. If peers caching the chunk have no enough upload bandwidth, the requests are rejected. We apply the loss network model [9] to calculate the chunk loss probability in the P2P VoD system.

Let $\mathbf{n_m} = \{n_{m,j}\}_{c_j \in \mathcal{C}}$ denote the vector of request numbers for different chunks being served concurrently in ISP $m$.

The loss probability for chunk $j$ in ISP $m$, $L_{m,j}$, can be calculated in the following: the requests under service experience a delay of 1 unit time (service time). The loss requests experience a delay of 0. The average delay that chunk requests experienced is $D_{m,j} = (1 - L_{m,j}) \cdot 1 + L_{m,j} \cdot 0 = (1 - L_{m,j})$. Upon applying Little's law to the VoD system (with respect to chunk $j$), we obtain $\nu_{m,j} D_{m,j} = \mathbf{E}[n_{m,j}]$, which yields

$$L_{m,j} = 1 - \frac{\mathbf{E}[n_{m,j}]}{\nu_{m,j}}.$$

Hence, the problem of obtaining the loss probability $L_{m,j}$ becomes deriving $\mathbf{E}[n_{m,j}]$. We take the 1-point approximate algorithm, using $n_{m,j}^*$, which is the element of $\mathbf{n_m^*}$, the state having the maximum probability, as a surrogate of $\mathbf{E}[n_{m,j}]$ []. Relax integer vector $\mathbf{n_m}$ using a real vector $\mathbf{x_m}$. $\mathbf{n_m^*}$ satisfies the following optimization problem:

$$\max \sum_{j=1}^{J} x_{m,j}log\nu_{m,j} - x_{m,j}logx_{m,j} + x_{m,j}$$
$$\text{over } \forall \mathcal{A} \subseteq \mathcal{C}, \sum_{c_j \in \mathcal{A}} x_{m,j} \leq U_m \cdot \sum_{i:\mathcal{A}\cap s_i \neq \emptyset} N_m^{(i)}$$
$$\mathbf{x_m} \geq 0$$

The corresponding Lagrangian is:

$$\begin{aligned} L(\mathbf{x_m}, \epsilon) &= \sum_{j=1}^{J}(x_{m,j}log\nu_{m,j} - x_{m,j}logx_{m,j} + x_{m,j}) \\ &+ \sum_{\mathcal{A}\subseteq\mathcal{C}} \epsilon_{\mathcal{A}} \cdot (U_m \cdot \sum_{i:\mathcal{A}\cap s_i \neq \emptyset} N_m^{(i)} - \sum_{j=1}^{M} x_{m,j}) \\ &= \sum_{j=1}^{J} x_{m,j} + \sum_{j=1}^{J} x_{m,j}(log\nu_{m,j} - logx_{m,j} \\ &- \sum_{c_j\in\mathcal{A},\mathcal{A}\subseteq\mathcal{C}} \epsilon_{\mathcal{A}}) + \sum_{\mathcal{A}\subseteq\mathcal{C}} \epsilon_{\mathcal{A}} \cdot U_m \cdot \sum_{i:\mathcal{A}\cap s_i\neq\emptyset} N_m^{(i)} \end{aligned}$$

The KKT conditions for this convex optimization problem are:

$$\forall \mathcal{A} \subseteq \mathcal{C}, \sum_{c_j\in\mathcal{A}} x_{m,j} \leq U_m \cdot \sum_{i:\mathcal{A}\cap s_i\neq\emptyset} N_m^{(i)} \quad (1)$$

$$\epsilon_{\mathcal{A}} \geq 0 \quad (2)$$

$$\forall \mathcal{A} \subseteq \mathcal{C}, \epsilon_{\mathcal{A}} \cdot (U_m \cdot \sum_{i:\mathcal{A}\cap s_i\neq\emptyset} N_m^{(i)} - \sum_{c_j\in\mathcal{A}} x_{m,j}) = 0 \quad (3)$$

$$x_{m,j} = \nu_{m,j} \cdot \exp(-\sum_{c_j\in\mathcal{A},\mathcal{A}\subseteq\mathcal{C}} \epsilon_{\mathcal{A}}) \quad (4)$$

When we get $\mathbf{x_m}$ from the above KKT conditions, we can calculate the system average chunk loss rate as:

$$L_m = \frac{\sum_{j=1}^{J}(1 - \frac{x_{m,j}}{\nu_{m,j}}) \cdot \nu_{m,j}}{\nu_m} = 1 - \frac{\sum_{j=1}^{J} x_{m,j}}{\nu_m} \quad (5)$$

### *B. Maximum Bipartite Flow Map of the KKT Conditions*

The total served chunk requests are $\sum_{j=1}^{J} x_{m,j}$, which is the sum of the solutions of the KKT conditions. The number of functions in KKT conditions grows exponentially with the number of chunks, which makes it computationally complex to solve the KKT conditions. We prove that the served request rates got from solutions of the KKT conditions can be mapped into the maximum bipartite flow in the corresponding bipartite graph (Fig. 1).

We first present the corresponding bipartite graph (Fig. 1) along with the source node $s$ and the destination node $t$, the bipartite graph has two sets of nodes $\Theta$ and $\mathcal{C}$ with edges directed from $\Theta$ to $\mathcal{C}$. The left set of nodes, $\Theta$, represents peers with different cache states, the right set of nodes, $\mathcal{C}$, represents different chunks. The edges directed from node $s_i$, representing peers with cache state $s_i$ in $\Theta$, to nodes in $\mathcal{C}$ represent the flow of upload bandwidth serving requests for chunks. The edges from source $s$ to any node in set $\Theta$ represent the peers' cache distribution and have the capacity, $N_m^{(i)} \cdot U_m$, for cache state $s_i$. The edges from $s_i \in \Theta$ to $c_j \in \mathcal{C}, c_j \in s_i$ have the capacity, $N_m^{(i)} \cdot U_m$, which can not exceed the total

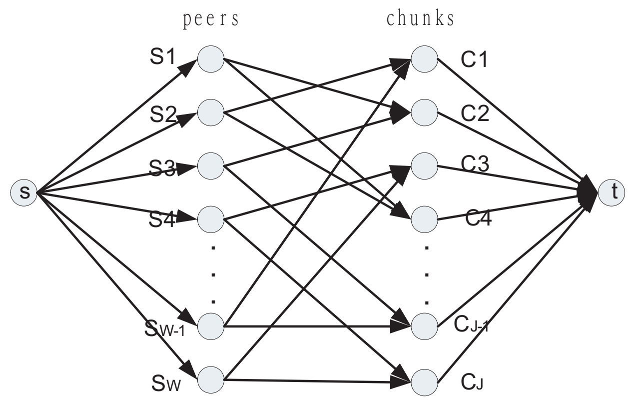


Figure 1. Corresponding Bipartite Graph.

upload bandwidth of $s_i$. The edges from any nodes in $\mathcal{C}$ to the destination $t$ represent the request rates for chunks, having a capacity of $\nu_{m,j}$.

**Theorem 1.** *The total served request rates, $\sum_{j=1}^{J} x_{m,j}$, obtained from the KKT conditions is the maximum bipartite flow of the corresponding bipartite graph (Fig. 1).*

*Proof:* We first construct a cutset that equals to the total served request rates satisfying the KKT conditions. We prove that this cutset is the minimum cut. Applying the min-cut-max-flow theorem, we can prove that the total served request rates is the maximum bipartite flow.

$x_{m,j}, 1 \leq j \leq J$ denote the solutions of the KKT conditions for the served request numbers. We can divide the $x_{m,j}$'s into two classes according whether $x_{m,j}$ is equal to $\nu_{m,j}$ or smaller than $\nu_{m,j}$. $\mathcal{C}_1 = \{c_j | x_{m,j} = \nu_{m,j}, 1 \leq j \leq J\}$, $\mathcal{C}_2 = \{c_j | x_{m,j} < \nu_{m,j}, 1 \leq j \leq J\}$. The peers' upload bandwidth for the chunks in set $\mathcal{C}_2$ is not enough. From the KKT conditions, for all $\mathcal{A}$, when $\mathcal{A}$ includes $c_j \in \mathcal{C}_1$, $\epsilon_{\mathcal{A}} = 0$; for $\mathcal{A} = \mathcal{C}_2$, as the peers' total upload bandwidth is not enough, we have $\sum_{c_j\in\mathcal{C}_2} x_{m,j} = U_m \cdot \sum_{i:s_i\cap\mathcal{C}_2\neq\emptyset} N_m^{(i)}$. According to KKT condition (3), we have $\epsilon_{\mathcal{C}_2} > 0$. Hence, it is easy to verify that the cutset from the set including source node $s$, all nodes in set $\mathcal{C}_2$, nodes in set $\Theta$ that have connections with nodes in set $\mathcal{C}_2$ to the residual set of the graph is $\sum_{c_j\in\mathcal{C}_1} \nu_{m,j} + U_m \cdot \sum_{i:s_i\cap\mathcal{C}_2\neq\emptyset} N_m^{(i)}$, equal to $\sum_{j=1}^{J} x_{m,j}$. In the following step, we prove that this cutset is the minimum cut.

On one hand, when the set including source node $s$ contains a node $C_j$ from $\mathcal{C}_1$, as the capacity of edges from node $C_j$ to nodes in $\Theta$ is equal to that of the edge from source $s$ to node $C_j$, the resulted cutset is no smaller than that of excluding the node from $\mathcal{C}_1$. When all nodes in $\Theta$ having connections with $C_j$ are also included in the set. As a result, the total capacity of cutset is increased by $U_m \cdot \sum_{s_i:c_j\in s_i} N_m^{(i)} - nu_{m,j} \geq 0, c_j \in \mathcal{C}_1$ according to KKT condition (1). On the other hand, when the set including source node $s$ excludes a node $C_k$ from $\mathcal{C}_2$, the cutset is increased by $\nu_{m,k}$. As a node $S_i$ in $\Theta$ having connections with $C_k$ is also excluded from the set including source node $s$, the cutset will be changed by $\sum_{j:j\in S_i} \nu_{m,j} -$

$U_m N_m^{(i)} \geq 0$. Hence, the total capacity will be increased.

The cutset with capacity $\sum_{j=1}^{J} x_{m,j}$ is the minimum cut. Applying the min-cut-max-flow theorem, we conclude that the total served request rates is the maximum bipartite flow.

## IV. Optimal Cache Condition and Optimal Chunk Request Routing

The model framework can use the maximum bipartite flow algorithm to solve the average chunk request loss probability under a specific chunk request rate distribution and a specific cache distribution. We are interested in the chunk request loss probability under the optimal cache distribution. We first state the optimal cache condition and the concrete optimal cache placement strategy. We will analyze the chunk request loss probability under the optimal cache.

### A. Optimal Cache Condition in P2P VoD system

The ISP $m$'s work load, $\eta_m$, is the ratio of the total number of chunk requests to the peers' total upload bandwidth:

$$\eta_m = \frac{\nu_m}{N_m U_m}$$

The optimal cache strategy in P2P VoD system makes the chunk requests can be served when the upload bandwidth is available. Hence, under the optimal cache, the achievable chunk request loss probability is related to the ISP's work load:

$$L_m = max\{0, 1 - \frac{1}{\eta_m}\}$$

**Lemma 1.** The optimal cache distribution should satisfy the following inequality:

$$\forall \mathcal{A} \subseteq \mathcal{C}, \sum_{c_j \in \mathcal{A}} x_{m,j} \leq \eta_m \cdot U_m \cdot \sum_{i: \mathcal{A} \cap s_i \neq \emptyset} N_m^{(i)}$$

*proof:* When $N_m^{(i)}$'s satisfy $\forall \mathcal{A} \subseteq \mathcal{C}, \sum_{c_j \in \mathcal{A}} x_{m,j} \leq \eta_m \cdot U_m \cdot \sum_{i: \mathcal{A} \cap s_i \neq \emptyset} N_m^{(i)}$, $x_{m,j} = min\{\nu_{m,j}, \frac{\nu_{m,j}}{\eta_m}\}, 1 \leq j \leq J$ is the solutions of the KKT conditions. The average chunk request loss probability achieved is $max\{1 - \frac{1}{\eta_m}, 0\}$

Lemma 1 has stated two cache placement strategies which satisfy the above inequality.

**Lemma 2.** *With different peer upload bandwidth allocation strategy, the corresponding optimal cache placement strategy is different. The following two cache placement strategies are both optimal under the corresponding peer upload bandwidth allocation strategy. Cache placement strategy 1: the proportion of cache state $s_i$ is $\gamma_i = \frac{\sum_{c_j \in s_i} \nu_{m,j}}{C_{J-1}^{B-1} \nu_m}$, the corresponding peers' upload bandwidth allocation among cached chunks is proportional to the chunk request rate; Cache placement strategy 2: the second is that the proportion of peers caching chunk $c_j$ is $\rho_j = \frac{\nu_{m,j}}{\nu_m}$. Peers allocating their upload bandwidth uniformly among cached chunks can achieve the optimal performance.*

*proof:* Let's consider the average request loss probability $L_m$ in ISP $m$. The total peer upload bandwidth in ISP $m$ is $N_m \cdot U_m$. The minimum average request loss probability can be reached is $max\{1 - \frac{N_m \cdot U_m}{\nu_m}, 0\}$. In cache placement strategy 1, the bandwidth used to serve chunk $c_j$ can be calculated as $N_m U_m \sum_{s_i : c_j \in s_i} \frac{\sum_{c_j \in s_i} \nu_{m,j}}{C_{J-1}^{B-1} \cdot \nu_m} \cdot \frac{\nu_{m,j}}{\sum_{c_j \in s_i} \nu_{m,j}} = \frac{\nu_{m,j} U_m N_m}{\nu_m}$. Thus the loss probability of requests for chunk $c_j$ is $min\{1 - \frac{N_m \cdot U_m}{\nu_m}, 0\}$. All chunks have the same loss probability $max\{1 - \frac{N_m \cdot U_m}{\nu_m}, 0\}$, which is the minimum average request loss probability. Thus cache placement strategy 1 is optimal. In cache placement strategy 2, the bandwidth used to serve chunk $c_j$ is $\frac{\nu_{m,j}}{\nu_m} \cdot B \cdot N_m \cdot \frac{U_m}{B} = \frac{\nu_{m,j} U_m N_m}{\nu_m}$. The loss probability of requests for chunk $c_j$ is $max\{1 - \frac{N_m \cdot U_m}{\nu_m}, 0\}$. All chunks have the same loss probability $max\{1 - \frac{N_m \cdot U_m}{\nu_m}, 0\}$, which is the minimum average request loss probability.

### B. Analysis of LRU Cache Replacement Strategy

(I intend to show that LRU replacement strategy in VoD system can achieve the optimal cache distribution. The probability $Pr$[users download chunk $j$ at time $n+1 | s_n^j = 0] = (1 - \phi) \cdot \pi_j$ seems not correct. As when given $s_n^j = 0$, the probability that users download chunk $j$ should be larger than $(1 - \phi) \cdot \pi_j$, as some chunks cached in the peer will not be downloaded. I am still thinking over this problem.)

In this section, we theoretically prove that the Least Recently Used (LRU) cache replacement strategy achieves the optimal cache distribution at its equilibrium state. We assume the distribution of chunk popularity is $(\pi_1, \pi_2, \ldots, \pi_J)$. We say a chunk is in different states when it is cached at different positions in a peer. We will first analyze a chunk's equilibrium states in one peer's cache.

Let us consider chunk $j$'s position in a peer's cache $n$ time unit after the peer starts playing video, which is denoted by $s_n^j$, under LRU algorithm. We assume users watching videos have a probability $\phi$ of taking backward behavior. Hence, the probability of normally playing or forwarding is $1 - \phi$. When users take backward behavior, they will watch the chunks that they have watched not long before and are cached in the peer's local cache. Users do not need to download new chunks and the watched chunk's position will become 1 in the cache. We assume when a user backwards, the selected positions have a uniformly distribution. Hence, the cached chunk will be randomly uniformly played. When users play normally or forward, a new chunk will be played and cached at position 1 in the cache. The chunk in the last position of cache, position $B$, will be evicted. Positions of all other cached chunks will increase by 1. Given $s_n^j$, we can derive the probability for chunk $j$'s position at time $n+1$, $s_{n+1}^j$.

When $2 \leq b \leq B$,

$$\begin{aligned} & Pr[s_{n+1}^j = b | s_n^j] = \\ & Pr[c_j\text{'s position increases by } 1 | s_n^j = b-1] \cdot Pr[s_n^j = b-1] \\ & + Pr[c_j\text{'s position does not change} | s_n^j = b] \cdot Pr[s_n^j = b] \end{aligned}$$

The event that chunk $j$'s position increases by 1 when $s_n^j = b - 1$ can be divided into two disjoint events: one is that the peer plays a new chunk; the other is that the peer plays a chunk cached at positions behind $b-1$:

$$Pr[c_j\text{'s position increases by } 1|s_n^j = b-1] = (1-\phi) + \phi \cdot \frac{B-b+1}{B}$$

The event that chunk $j$'s position does not change when $s_n^j = b$ happens when the peer backwards and plays a chunk cached at positions ahead of $b$:

$$Pr[c_j\text{'s position does not change}|s_n^j = b] = \phi \cdot \frac{b-1}{B}$$

Hence,

$$Pr[s_{n+1}^j = b|s_n^j] = (1+\phi \cdot \frac{1-b}{B}) \cdot Pr[s_n^j = b-1] + \phi \cdot \frac{b-1}{B} \cdot Pr[s_n^j = b] \quad (6)$$

When $b=1$,

$$Pr[s_{n+1}^j = 1|s_n^j] = Pr[\text{users download } c_j \text{ at time } n+1|s_n^j = 0] \cdot Pr[s_n^j = 0] + Pr[\text{users backward and replay } c_j|s_n^j \neq 0] \cdot Pr[s_n^j \neq 0]$$

$Pr[s_n^j = 0]$ means chunk $j$ is not cached in the peer's local cache.

$$Pr[\text{users download } c_j \text{ at time } n+1|s_n^j = 0] = (1-\phi) \cdot \pi_j$$

$$Pr[\text{users backward and replay } c_j|s_n^j \neq 0] = \phi \cdot \frac{1}{B}$$

Hence,

$$Pr[s_{n+1}^j = 1|s_n^j] = (1-\phi)\pi_j \cdot Pr[s_n^j = 0] + \phi\frac{1}{B} \cdot Pr[s_n^j \neq 0] \quad (7)$$

Equations (7) and (8) show that the next position of chunk $j$ is only related to the previous position of chunk $j$, we could model the change of chunk $j$'s positions as a Markov Chain. We use state $b, (1 \leq b \leq B)$ to denote chunk $j$'s position is at cell $b$ of the peer's cache. State 0 denotes chunk $j$ is not in the peer's cache. Hence, $s_{n+1}^j$ denotes the state of chunk $j$ at time slot $n+1$.

We analyze the stationary states that a peer caches chunk $j$, which are the probability distribution for chunk $j$'s positions at a peer's cache. Let $s^j$ denote the stationary state of chunk $j$ when $n$ increases to infinity. We have, for $2 \leq b \leq B$,

$$Pr[s^j = b] = (1+\phi \cdot \frac{1-b}{B}) \cdot Pr[s^j = b-1] + \phi \cdot \frac{b-1}{B} \cdot Pr[s^j = b] \quad (8)$$

for $b=1$,

$$Pr[s^j = 1] = (1-\phi)\pi_j \cdot Pr[s^j = 0] + \phi\tfrac{1}{B} \cdot Pr[s^j \neq 0] \quad (9)$$

Hence, $Pr[s^j = 1] = Pr[s^j = 2] = \ldots = Pr[s^j = B] = \frac{\pi_j}{1+\pi_j B}$. $Pr[s^j = 0] = \frac{1}{1+\pi_j B}$

The probability that a peer cache chunk $j$ is $\sum_{b=1}^{B} Pr[s^j = b] = \frac{\pi_j B}{1+\pi_j B}$ under the stationary states of LRU replacement strategy.

### C. Optimal Chunk Request Routing

Let us now consider how to achieve the minimum average request loss probability for VoD system under the optimal cache. We have the average request loss probability in ISP $m$:

$$L_m = max\{1 - \frac{U_m \cdot N_m}{\sum_{j=1}^{J} \nu_{m,j}}, 0\}$$

The average chunk loss rate for the VoD system is:

$$L = \frac{\sum_{m=1}^{M} L_m \cdot \nu_m}{\sum_{m=1}^{M} \nu_m} = \frac{\sum_{m=1}^{M} max\{\nu_m - U_m \cdot N_m, 0\}}{\sum_{m=1}^{M} \nu_m}$$

The minimum system chunk loss rate can be achieved through the following optimization problem:

$$\begin{aligned} min \quad & \frac{\sum_{m=1}^{M} max\{\nu_m - U_m \cdot N_m, 0\}}{\sum_{m=1}^{M} \nu_m} \\ over \quad & \nu_m = \sum_{l=1}^{M} a_{lm} \cdot r_l \\ & \sum_{m=1}^{M} a_{lm} = 1, 1 \leq l \leq M \end{aligned}$$

Let us consider the optimization problem in two cases:

1) The peers' total upload bandwidth is larger than the demand of chunk requests, $\sum_{m=1}^{M} U_m N_m \geq \sum_{m=1}^{M} \nu_m$.

In this case, there exist $a_{lm}$, which satisfy $\nu_m = \sum_{l=1}^{M} a_{lm} \cdot r_l \leq U_m N_m$, for $1 \leq m \leq M$.The objective function can take minimum value $L = 0$. The value of $a'_{lm}s$ can be obtained through the following inequalities:

$$\sum_{l=1}^{M} a_{lm} r_l \leq U_m N_m, 1 \leq m \leq M$$
$$\sum_{m=1}^{M} a_{lm} = 1, 1 \leq l \leq M$$

2) The peers' total upload bandwidth is smaller than the demand of chunk requests, $\sum_{m=1}^{M} U_m N_m < \sum_{m=1}^{M} \nu_m$.

In this case, we have

$$\begin{aligned} & \frac{\sum_{m=1}^{M} max\{\nu_m - U_m \cdot N_m, 0\}}{\sum_{m=1}^{M} \nu_m} \geq \\ & \frac{max\{\sum_{m=1}^{M} (\nu_m - U_m \cdot N_m), 0\}}{\sum_{m=1}^{M} \nu_m} = \frac{\sum_{m=1}^{M} (\nu_m - U_m \cdot N_m)}{\sum_{m=1}^{M} \nu_m} \\ & = 1 - \frac{\sum_{m=1}^{M} U_m N_m}{\sum_{m=1}^{M} \nu_m} \end{aligned}$$

The minimum system chunk request loss probability is $L = 1 - \frac{\sum_{m=1}^{M} U_m N_m}{\sum_{m=1}^{M} \nu_m}$. To obtain this minimum system chunk request loss probability, we just need $\frac{\sum_{m=1}^{M} max\{\nu_m - U_m \cdot N_m, 0\}}{\sum_{m=1}^{M} \nu_m} = \frac{max\{\sum_{m=1}^{M} (\nu_m - U_m \cdot N_m), 0\}}{\sum_{m=1}^{M} \nu_m}$. Hence, we have,

$$\nu_m = \sum_{l=1}^{M} a_{lm} r_l \geq U_m N_m, 1 \leq m \leq M$$
$$\sum_{m=1}^{M} a_{lm} = 1, 1 \leq l \leq M$$

## V. Cross-ISP traffic and Performance Relationship

### A. Cross-ISP traffic under minimum chunk loss probability

We first introduce an ISP-aware algorithm to minimize chunk loss probability and reduce unnecessary cross-ISP traffic.

The tracker sorts the ISPs according to the value of $I_m = (U_m \cdot N_m - r_m)$, which has a positive value when peer resource in ISP $m$ is larger than the chunk requests, a negative value when peer resource in ISP $m$ is smaller than the chunk requests.

The tracker sets the value of $a_{ml}$'s as follows:

For ISPs with $I_m = U_m \cdot N_m - r_m \geq 0$, $a_{mm} = 1$, $a_{ml} = 0$ for $1 \leq l \leq M, l \neq m$.

For ISPs with $I_m = U_m \cdot N_m - r_m < 0$, $a_{mm} = \frac{U_m \cdot N_m}{r_m}$, $a_{ml} = \frac{I_l}{\sum_{k,I_k>0} I_k} \frac{r_m - U_m N_m}{r_m}$ for ISP $l$ with $I_l = U_l N_l - r_l \geq 0$, $a_{ml} = 0$ for ISP $l$ with $I_l = U_l N_l - r_l < 0$.

Let each peer in ISP $m$ keep a proportion of $a_{ml}$ neighbors from ISP $l$, $1 \leq l \leq M$. Peers refresh their neighbor lists to keep peers in their neighbor lists having chunks they want to download. They randomly send their chunk requests to their neighbors. As a result, according to Chernoff Bound, with high probability, $a_{ml}$ proportion of chunk requests will be routed from ISP $m$ to ISP $l$.

When $I_m \geq 0$, the total chunk requests that ISP $m$ needs to serve are:

$$\nu_m = r_m + \sum_{k,I_k<0} \frac{I_m}{\sum_{t,I_t>0} I_t} \frac{r_k - U_k N_k}{r_k} r_k$$
$$= r_m - \sum_{k,I_k<0} \frac{I_m I_k}{\sum_{t,I_t>0} I_t}$$

When $I_m < 0$,

$$\nu_m = \frac{U_m \cdot N_m}{r_m} r_m$$
$$= U_m N_m$$

It is easy to verify that the chunk request distribution under this algorithm satisfies the optimal chunk request routing requirement. Hence, the ISP-aware algorithm achieves the minimum chunk loss probability with high probability.

For ISP with $I_m < 0$, $L_m = 0$. For ISP with $I_m \geq 0$, $L_m = max\{1 - \frac{U_m N_m}{\nu_m}, 0\}$.

Let $T_m^i$ be cross-ISP traffic from other ISPs to ISP $m$.

$$T_m^i = \sum_{k,I_k \geq 0} a_{mk} \cdot r_m \cdot (1 - L_k).$$

The total generated cross-ISP traffic can be calculated as follows:

$$T = \sum_{k,I_k<0} T_k^i$$

### B. Relationship between chunk loss probability and cross-ISP traffic

In this part, we study how the volume of cross-ISP traffic will affect the P2P VoD streaming system's chunk loss probabilities. We still study it under the optimal peers' cache distribution. This implies that we just need to consider the limitation of total peers' upload bandwidth.

We change the proportion of chunk requests routed to other ISPs to increase or reduce the inter-ISP traffic. Then, we see how the chunk loss probability changes with the cross-ISP traffic.We analyze cross-ISP traffic impact on chunk loss probability.

We change the cross-ISP traffic through changing the value of $a_{ml}$'s. For ISPs with $I_m \geq 0$, let $a_{mm} = \alpha$, $a_{ml} = (1-\alpha) \cdot \frac{N_l}{N - N_m}$, $1 \leq l \leq M, l \neq m$. For ISPs with $I_m < 0$, let $a_{mm} = \beta \cdot \frac{U_m \cdot N_m}{r_m}$, $a_{ml} = \frac{I_l}{\sum_{k,I_k>0}(I_k)} \frac{r_m - U_m N_m}{r_m} + (1-\beta) \cdot \frac{U_m \cdot N_m}{r_m} \cdot \frac{N_l}{N - N_m}$ for ISP $l$ with $I_l \geq 0$. $a_{ml} = (1-\beta) \cdot \frac{U_m \cdot N_m}{r_m} \cdot \frac{N_l}{N - N_m}$ for ISP $l$ with $I_l < 0$.

When $\beta = \alpha \in (0, 1)$, the chunk requests routed to other ISPs increase. For ISPs with $I_m \geq 0$, we have

$$\nu_m = \textstyle\sum_{l=1}^{M} a_{lm} r_l$$
$$= \alpha r_m + \textstyle\sum_{k,I_k \geq 0} (1-\alpha) \cdot \frac{N_m}{N - N_k} r_k +$$
$$\textstyle\sum_{k,I_k<0} [\frac{I_m}{\sum_{t,I_t>0} I_t} \frac{r_k - U_k N_k}{r_k} + (1-\beta) \cdot \frac{U_k \cdot N_k}{r_k} \cdot \frac{N_m}{N - N_k}] r_k$$

For ISPs with $I_m < 0$, we have

$$\nu_m = \sum_{l=1}^{M} a_{lm} r_l$$
$$= \beta \cdot \frac{U_m \cdot N_m}{r_m} r_m + \sum_{k,I_k \geq 0} (1-\alpha) \cdot \frac{N_m}{N - N_k} r_k +$$
$$\sum_{k,I_k<0} [(1-\beta) \cdot \frac{U_k \cdot N_k}{r_k} \cdot \frac{N_m}{N - N_k}] r_k$$

The cross-ISP traffic can be calculated as:

$$T_m^i = \sum_{l=1, l \neq m}^{M} a_{ml} r_m (1 - L_l)$$

When $\alpha = 1$, $\beta \in (1, \frac{r_m}{U_m \cdot N_m})$, the cross-ISP chunk requests are reduced. For ISPs with $I_m \geq 0$, $a_{mm} = 1$, $a_{ml} = 0$, for ISPs with $I_m < 0$, $a_{mm} = \beta \cdot \frac{U_m \cdot N_m}{r_m}$, $a_{ml} = \frac{I_l}{\sum_{k,I_k>0}(I_k)} \frac{r_m - U_m N_m}{r_m} + (1-\beta) \cdot \frac{U_m \cdot N_m}{r_m} \cdot \frac{N_l}{N - N_m}$ for ISP $l$ with $I_l \geq 0$, $a_{ml} = 0$ for ISP $l$ with $I_l < 0$. For ISPs with $I_m \geq 0$, we have

$$\nu_m = \textstyle\sum_{l=1}^{M} a_{lm} r_l$$
$$= r_m + \textstyle\sum_{k,I_k<0} [\frac{I_m}{\sum_{t,I_t>0} I_t} \frac{r_k - U_k N_k}{r_k} + (1-\beta) \frac{U_k N_k}{r_k} \frac{N_m}{N - N_k}] r_k$$
$$= r_m + \textstyle\sum_{k,I_k<0} [\frac{I_m}{\sum_{t,I_t>0} I_t} (r_k - U_k N_k) + (1-\beta) U_k N_k \frac{N_m}{N - N_k}]$$

For ISPs with $I_m < 0$, we have

$$\nu_m = \beta \cdot \frac{U_m \cdot N_m}{r_m} r_m$$
$$= \beta \cdot U_m \cdot N_m$$

With $\nu_m$, we can get the chunk loss probability in each ISP, $L_m$. Hence, we can calculate the cross-ISP traffic using the following formula:

$$T_m^i = \sum_{l=1, l \neq m}^{M} a_{ml} r_m (1 - L_l)$$

$$T = \sum_{m=1}^{M} T_m^i$$

## VI. Performance Evaluation

We carry out numerical analyses using parameters driven from the empirical data in the real-world. We simulate 10 ISPs. The total number of concurrent users over the system is 100000. The users distribute in ISPs according to the probability distribution $p_{isp}(m) = \frac{(M-m+1)^{\beta}}{\sum_{m=1}^{M}(M-m+1)^{\beta}}$. The average upload bandwidth of each ISP equals to the playingback rate. The total number of different chunks shared in the system is 1000. Every peer has a cache of 50 chunks. The chunk popularity in the system is simulated using the Zipf-Mandelbrot distribution $\pi(j) = \frac{\frac{1}{(j+q)^{\alpha}}}{\sum_{j=1}^{J} \frac{1}{(j+q)^{\alpha}}}$. The number of peer neighbors is $d = 30$.

The change of chunk loss rate with the number of inter-ISP connections is presented under both optimal cache placement strategy and unoptimal cache placement strategy. Fig. 2 (a) shows that under optimal cache, the chunk loss rate of ISPs with fewer peers (ISP10) increases as peers' inter-ISP connections increases. ISPs with more peers (ISP1) have abundant upload bandwidth as peers' inter-ISP connections increases. This is due to the reason that as peers in different ISPs have the same number of inter-ISP connections, the ISP with more peers route more chunk requests to other ISPs than those routed from other ISPs into it. This implies the peer in ISP with fewer peers should keep more inter-ISP connections in ISP-aware VoD design. Fig. 2 (b) shows the chunk loss rate and inter-ISP connection relationship under an unoptimal cache placement: all chunks have the same distribution probability regardless of the chunk popularity. We see that the chunk loss rate increases largely compared with that under optimal cache placement. In this case, the impact of inter-ISP connections on performance is not obvious.

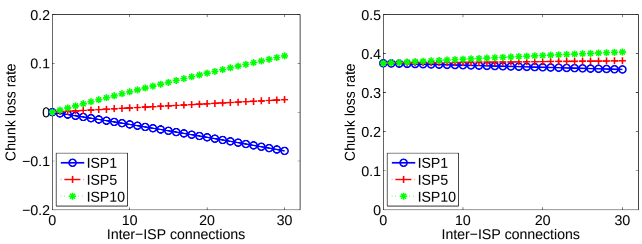


Figure 2. Chunk loss rate and inter-ISP connections relationship under optimal and unoptimal peer cache distribution.

## VII. Conclusions

This paper targets theoretical study of relationship between controlled inter-ISP connections and system performance in ISP-aware P2P VoD system. We apply the stochastic loss network model to analyze the problem, map the solutions to the corresponding maximum bipartite flow and design an effective algorithm to solve the maximum bipartite flow. We not only settle the general peer cache case, but also obtain the analytical results for the optimal peer cache case.

## References


[1] *PPLive*, http://www.pplive.tv.

[2] *UUSee*, http://www.uusee.com.

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

[4] W. Huang, C. Wu, and F. C. M. Lau, "The Performance and Locality Tradeoff in BitTorrent-like P2P File-sharing Systems," in *Proc. of IEEE ICC*, May 2010.

[5] F. Picconi and L. Massoulie, "ISP Friend or Foe? Making P2P Live Streaming ISP-Aware," in *Proc. of IEEE ICDCS*, June 2009.

[6] J. Wang, C. Huang, and J. Li, "On ISP-friendly Rate Allocation for Peer-assisted VoD," in *Proc. of ACM Multimedia 2008*, 2008.

[7] D. P. Heyman and M. J. Sobel, *Stochastic Models in Operations Research: Stochastic Processes and Operating Characteristics*. Courier Dover, 2004.

[8] F. Kelly, "Loss networks," *The Annals of Applied Probability*, vol. 1, no. 3, pp. 319–378, Aug. 1991.

[9] B. R. Tan and L. Massoulié, "Optimal Content Placement for Peer-to-Peer Video-on-Demand Systems," in *Proc. of IEEE INFOCOM*, April 2011.