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. System 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 peer average upload bandwidth in ISP m is U_m. The VoD system supplies multiple video channels. The videos are divided into chunks for storage and advertising to neighbors which parts of the video a peer caches []. 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 can cache B chunks. $s_i = \{c_1^{(i)}, c_2^{(i)}, ..., c_B^{(i)}\}$ denotes the cache state i. Let Θ be the set of all different cache states of peers, $W = |\Theta|$. The servers have cached all chunks.

A. Peers' Cache State Distribution

In the VoD system, a peer contributes a storage of size B to cache chunks. The state of a peer's cache is $s_i = \{c_1^{(i)}, c_2^{(i)}, \ldots, c_B^{(i)}\}, 1 \le i \le W$. $N_m^{(i)}$ 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 γ_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 ϕ denote the probability that a peer replays cached chunks and does not need to download new chunks. We define chunk j's popularity as the probability that peers in the VoD system are playing chunk j. $(\pi_1, \pi_2, \pi_3, \ldots, \pi_J)$ denotes the chunk popularity distribution. A request for a specific 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 [1], 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 when a peer requests for chunk j, the request can be served by its own cache when it has cached chunk j, no downloading is necessary, and 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 $r_m = \sum_{j=1}^{J} r_{m,j} = \lambda_m \sum_{j=1}^{J} \pi_j \cdot (1-\phi) = (1-\phi)\lambda_m$.

The chunk requests may be served by peers in other ISPs. Let a_{ml} denotes the proportion of chunk requests routed from ISP m to ISP l. The total 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:

$$\nu_m = \sum_{j=1}^{J} \nu_{m,j} = \sum_{l=1}^{M} a_{lm} \cdot r_l.$$

C. Chunk Loss Rate

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 rate of requests 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 average 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:

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

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

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

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.
Θ	the set of all possible cache states of peers.
W	the number of different cache states.
ρ_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.
ϕ	the probability that a peer replays its cached chunks.
$L_{m,j}$	the loss rate for chunk j in ISP m.
T_m	the cross-ISP traffic flowing into ISP m.
T	the total cross-ISP traffic in VoD system.

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

II. A MODEL FRAMEWORK FOR CHUNK LOSS IN 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 [2] [3]. 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. A peer sending 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 [3] to calculate the chunk loss rate 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 chunk loss probability for chunk j in ISP m, $L_{m,j}$, can be calculated as follows: 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 experience 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 in ISP m (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 $\mathbf{E}[n_{m,j}]$ []. Relaxing integer vector $\mathbf{n_m}$ using a real vector $\mathbf{x_m}$. $\mathbf{n^*_m}$ satisfies the following optimization problem []:

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

The corresponding Lagrangian is:

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

The KKT conditions for this convex optimization problem are:

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

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

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

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

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) with source node s the destination node t. The bipartite graph has two sets of nodes: chunk set $\mathcal{C}$ and peer cache state setΘ. The left set of nodes, $\mathcal{C}$, represent different chunks, the right set of nodes, Θ, represent different peer cache states. The edges directed from nodes in $\mathcal{C}$ to nodes in Θ represent requests for a chunk can be sent to those states caching the chunk for downloading service. The edges from source s to any node in set $\mathcal{C}$ represent the requests for different chunks. The corresponding edge capacity is the chunk request rate in

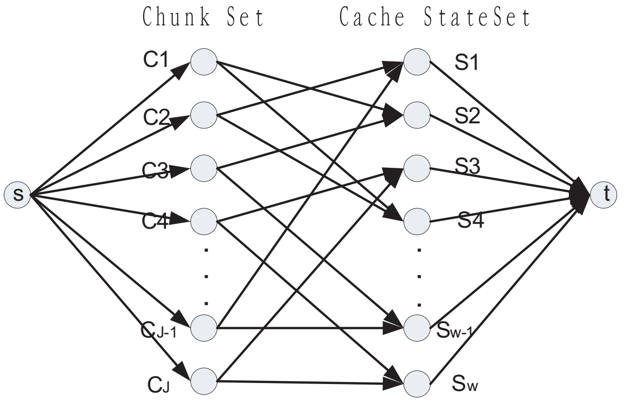

Figure 1. Corresponding Bipartite Graph.

an ISP. The edges from $c_j \in \mathcal{C}$ to $s_i \in \Theta, c_j \in s_i$ have the capacity, $\nu_{m,j}$, which can not exceed the total request rate served by ISP m. The edges from any node in Θ to the destination t represent cache states serving request for their caching chunks. The edge capacity is the total upload bandwidth of a cache state, which is $N_m^{(i)} \cdot U_m$.

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

Proof: We first construct a graph cut whose cutset equals to the total served request rate obtained from the KKT conditions. We prove that this cutset is the minimum cut. Applying the min-cut-max-flow theorem, we can show that the total served request rate is the maximum bipartite flow.

Let $x_{m,j}, 1 \le j \le J$ denote the solutions of the KKT conditions for the serving 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 $\nu_{m,j}$. $\mathcal{C}_1 = \{c_j | x_{m,j} = \nu_{m,j}, 1 \le j \le J\}$, $\mathcal{C}_2 = \{c_j | x_{m,j} < \nu_{m,j}, 1 \le j \le J\}$. 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 for chunks in $\mathcal{C}_2$, 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 Θ 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 Θ 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 Θ having connections with c_j are also included in the set. As a result, the total capacity of cutset is increased by $U_m \cdot \sum_{s_i: c_j \in s_i} N_m^{(i)} - \nu_{m,j} \ge 0, c_j \in \mathcal{C}_1$ 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 Θ having connections with c_k is also excluded from the set including source node s, the cutset will be changed by $\sum_{j: j \in s_i} \nu_{m,j} - U_m N_m^{(i)}$, which is larger than 0 as the peer upload bandwidth for requests of chunks in $\mathcal{C}_2$ is not enough. 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 have proved that the total served request rate is the maximum bipartite flow.

III. Optimal Cache Condition and Optimal Chunk Request Routing

The model framework can use the maximum bipartite flow algorithm to calculate the average chunk request loss rate under a specific cache state distribution and a specific chunk request rate distribution. We are especially interested in the chunk request loss rate under the optimal peer cache distribution. We first state the optimal peer cache condition. We propose a concrete optimal cache placement strategy and prove it satisfies the optimal cache condition. We also analyze the Least Recently Used (LRU) cache replacement strategy and show that it achieves close-to-optimal cache distribution in its stationary states. With the optimal peer cache, we analyze the chunk request routing among different ISPs.

A. Optimal Cache Condition in P2P VoD System

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

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

We have the following lemma for the optimal cache condition.

Lemma 1. When the number of peers with cache state s_i in ISP m, $N_m^{(i)}, 1 \le i \le W$, satisfies the following inequalities:

$$\forall \mathcal{A} \subseteq \mathcal{C}, \sum_{c_j \in \mathcal{A}} \nu_{m,j} \le \eta_m \cdot U_m \cdot \sum_{i: \mathcal{A} \cap s_i \neq \emptyset} N_m^{(i)} \qquad (6)$$

The system can achieve the chunk loss rate $L_m = max\{0, 1 - \frac{1}{\eta_m}\}$.

proof: When $N_m^{(i)}$'s satisfy (6), let $\epsilon_{\mathcal{A}} = 0$ for all $\mathcal{A} \subset \mathcal{C}$, $\epsilon_{\mathcal{A}} = max\{0, \ln \eta_m\}$ for $\mathcal{A} = \mathcal{C}$. It is easy to verify that $x_{m,j} = min\{\nu_{m,j}, \frac{\nu_{m,j}}{\eta_m}\}, 1 \le j \le J$ are the solutions of the KKT conditions. Hence, the average chunk request loss rate is $L_m = max\{0, 1 - \frac{1}{\eta_m}\}$.

Lemma 2 states a concrete optimal cache placement strategy.

Lemma 2. *One optimal cache placement strategy is to let chunk c_j be randomly cached in a proportion of $\rho_j = B \cdot \frac{\nu_{m,j}}{\nu_m} = B \cdot \pi_j$ peers.*

proof: We prove it by showing that the optimal strategy in Lemma 2 satisfies the optimal cache condition in Lemma 1.

We use the induction method.
(i) When $|\mathcal{A}| > J - B$, the intersection between any cache state $s_i, 1 \le i \le W$ and $\mathcal{A}$ will not be empty. Hence, the L.H.S of (6) equals to $\sum_{c_j \in \mathcal{A}} \nu_{m,j}$, which is smaller than the R.H.S of (6), $\eta_m \cdot U_m \cdot N_m = \nu_m$.
(ii) Let $\mathcal{A}_k$ denote the set $|\mathcal{A}| = k$. When $|\mathcal{A}| = J-B$, we use $\mathcal{A}_{J-B}$ in place of $\mathcal{A}$. Let $\bar{\mathcal{A}}_{J-B}$ be the complementary set of $\mathcal{A}_{J-B}$. The intersection between the cache state $s_i = \bar{\mathcal{A}}_{J-B}$ and $\mathcal{A}_{J-B}$ is empty.

$$\begin{aligned}
&\text{R.H.S of } (6) = \eta_m \cdot U_m(N_m - \sum_{i:s_i=\bar{\mathcal{A}}_{J-B}} N_m^{(i)}) \\
&= \eta_m \cdot U_m(N_m - N_m \cdot (\frac{B}{\nu_m})^B \prod_{c_j \in \bar{\mathcal{A}}_{J-B}} \nu_{m,j}) \\
&\ge \eta_m \cdot U_m(N_m - N_m \cdot (\frac{B}{\nu_m})^B \cdot (\frac{\sum_{c_j \in \bar{\mathcal{A}}_{J-B}} \nu_{m,j}}{B})^B) \\
&\ge \nu_m - (\sum_{c_j \in \bar{\mathcal{A}}_{J-B}} \nu_{m,j}) \cdot (\frac{\sum_{c_j \in \bar{\mathcal{A}}_{J-B}} \nu_{m,j}}{\nu_m})^{B-1} \\
&\ge \sum_{c_j \in \mathcal{A}_{J-B}} \nu_{m,j} = \text{L.H.S of } (6)
\end{aligned}$$

(iii) For $k \le J - B$, suppose $\forall \mathcal{A}_k$ satisfy (6). Let's see the case for any $\mathcal{A}_{k-1}$. Let us consider a specific $\mathcal{A}_{k-1}$, there are $(J-k+1)$ chunks $c_j \notin \mathcal{A}_{k-1}$. For any $c_j \notin \mathcal{A}_{k-1}$, we can construct a $\mathcal{A}_k = c_j \cup \mathcal{A}_{k-1}$. Hence, we get $(J-k+1)$ sets of $\mathcal{A}_k$. Let $\mathcal{A}_k^1, \mathcal{A}_k^2, \ldots, \mathcal{A}_k^{J-k+1}$ denote them. For each $\mathcal{A}_k$, we apply (6),

$$\sum_{c_j \in \mathcal{A}_k^t} \nu_{m,j} \le \eta_m \cdot U_m \cdot \sum_{i:\mathcal{A}_k^t \cap s_i \ne \emptyset} N_m^{(i)}, 1 \le t \le J-k+1$$

Sum up all the $(J-k+1)$ inequalities, we have,

$$\begin{aligned}
&\sum_{t=1}^{J-k+1} \sum_{c_j \in \mathcal{A}_k^t} \nu_{m,j} = (J-k+1) \sum_{c_j \in \mathcal{A}_{k-1}} \nu_{m,j} + \sum_{c_j \notin \mathcal{A}_{k-1}} \nu_{m,j} \\
&\le \sum_{t=1}^{J-k+1} \eta_m \cdot U_m \cdot \sum_{i:\mathcal{A}_k^t \cap s_i \ne \emptyset} N_m^{(i)} \\
&= (J-k+1)\eta_m \cdot U_m \sum_{i:\mathcal{A}_{k-1} \cap s_i \ne \emptyset} N_m^{(i)} \\
&+ \eta_m \cdot U_m \sum_{i:s_i \cap \mathcal{A}_{k-1} = \emptyset} N_m^{(i)}
\end{aligned}$$

Hence,

$$\begin{aligned}
&(J-k+1) \sum_{c_j \in \mathcal{A}_{k-1}} \nu_{m,j} + \nu_m - \sum_{c_j \in \mathcal{A}_{k-1}} \nu_{m,j} \\
&\le (J-k+1)\eta_m \cdot U_m \sum_{i:\mathcal{A}_{k-1} \cap s_i \ne \emptyset} N_m^{(i)} \\
&+ \eta_m \cdot U_m(N_m - \sum_{i:\mathcal{A}_{k-1} \cap s_i \ne \emptyset} N_m^{(i)})
\end{aligned}$$

We have,

$$\sum_{c_j \in \mathcal{A}_{k-1}} \nu_{m,j} \le \eta_m \cdot U_m \sum_{i:\mathcal{A}_{k-1} \cap s_i \ne \emptyset} N_m^{(i)}$$

Hence, we prove Lemma 2.

B. Optimality Property of LRU Cache Replacement Algorithm

With the optimal cache placement strategy, in this part, we theoretically prove that the system's stationary cache state distribution under Least Recently Used (LRU) cache replacement strategy is the optimal cache distribution proposed in Lemma 2.

Let us consider chunk j's position in a peer's cache n time unit after the peer starts playing videos, under LRU algorithm. We define a chunk's position in a peer's cache as the chunk's state, which is denoted by s_n^j. We first derive the state transition equations. Note that users watching videos have a probability ϕ of replaying videos. As users replay videos, they watch the chunks that they have watched not long before and are already cached in the peer's cache, users do not need to download new chunks and the replayed chunk's position will change to 1 in the cache. Hence, the probability of playing new contents is $1-\phi$. We assume when a user replays videos, the starting positions of replaying have a uniformly distribution among cached chunks. Hence, the cached chunk will be randomly uniformly played. When users play new contents, a new chunk will be downloaded 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 $2 \le b \le B$,

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

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 replays a chunk cached at positions behind $b-1$:

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

The event that chunk j's position does not change when $s_n^j = b$ happens when the peer replays 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,

$$\begin{aligned}
&Pr[s_{n+1}^j = b | s_n^j] = \\
&(1 + \phi \cdot \frac{1-b}{B}) \cdot Pr[s_n^j = b-1] + \phi \cdot \frac{b-1}{B} \cdot Pr[s_n^j = b]
\end{aligned} \tag{7}$$

Equation (8) shows that the next state of chunk j is only related to the previous state of chunk j, we can use Markov Chain to model the change of chunk j's states. 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. s_{n+1}^j denotes the state of chunk j at time slot $n+1$.

We analyze the stationary state distribution for a peer caching chunk j, which is 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 $2 \leq b \leq B$,

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

Hence, $Pr[s^j = 1] = Pr[s^j = 2] = \ldots = Pr[s^j = B]$, $Pr[s^j = 0] = 1 - \sum_{b=1}^{B} Pr[s^j = b] = 1 - B \cdot Pr[s^j = 1]$. $Pr[s^j = 1]$ is the probability that a peer is playing chunk j, which is the popularity of chunk j. Hence, $Pr[s^j = 1] = \pi_j$. The probability that a peer caches chunk j is $B \cdot Pr[s^j = 1] = B \cdot \pi_j$, which also means the proportion of peers caching chunk j is $\rho_j = B \cdot Pr[s^j = 1] = B \cdot \pi_j$. This is just the optimal cache distribution proposed in Lemma 2.

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:

$$\sum_{l=1}^{M} a_{lm} r_l \leq U_m N_m, 1 \leq m \leq M \qquad (9)$$
$$\sum_{m=1}^{M} a_{lm} = 1, 1 \leq l \leq M$$

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,

$$\nu_m = \sum_{l=1}^{M} a_{lm} r_l \geq U_m N_m, 1 \leq m \leq M \qquad (10)$$
$$\sum_{m=1}^{M} a_{lm} = 1, 1 \leq l \leq M$$

IV. CROSS-ISP TRAFFIC AND PERFORMANCE RELATIONSHIP

A. Necessary Cross-ISP Traffic for Minimum Chunk Loss Probability

We first propose an ISP-aware algorithm to reduce unnecessary cross-ISP traffic while minimizing chunk loss probability.

ISP-aware Chunk Request Routing Algorithm: (1) 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 its generated chunk requests, a negative value when peer resource in ISP m is smaller than its generated chunk requests. (2) The tracker sets the value of a_{ml}'s as follows: For ISPs with $I_m = U_m \cdot N_m - r_m \geq 0$, $a_{mm} = 1, a_{ml} = 0, 1 \leq l \leq M, l \neq m$. For ISPs with $I_m = U_m \cdot N_m - r_m < 0$, $a_{ml} = min\{\frac{\sum_{t, I_t > 0} I_t}{-\sum_{s, I_s < 0} I_s}, 1\}(\frac{r_m - N_m U_m}{r_m}) \cdot \frac{I_l}{\sum_{t, I_t > 0} I_t}$ for ISP l with $I_l = U_l N_l - r_l \geq 0, a_{ml} = 0$ for ISP l with $I_l = U_l N_l - r_l < 0, a_{mm} = 1 - \sum_{l=1, l \neq m}^{M} a_{ml} = 1 - min\{\frac{\sum_{t, I_t > 0} I_t}{-\sum_{s, I_s < 0} I_s}, 1\}(\frac{r_m - N_m U_m}{r_m}) = max\{1 - \frac{\sum_{t, I_t > 0} I_t}{-\sum_{s, I_s < 0} I_s}(\frac{r_m - N_m U_m}{r_m}), \frac{N_m U_m}{r_m}\}$. (3) Each peer in ISP m keeps a proportion of a_{ml} neighbors from ISP l, $1 \leq l \leq M$. Peers update 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. Hence, with probability a_{ml}, the chunk requests generated in ISP m will be routed to peers in ISP l.

We can thus calculate the total chunk requests that ISP m needs to serve under the ISP-aware Chunk Request Routing Algorithm. When $I_m \geq 0$, the total chunk requests that ISP

m needs to serve are:

$$\nu_m = r_m + \sum_{k,I_k<0} min\{\frac{\sum_{t,I_t>0} I_t}{-\sum_{s,I_s<0} I_s}, 1\} \frac{r_k - U_k N_k}{r_k} \frac{I_m}{\sum_{t,I_t>0} I_t} r_k$$
$$= r_m - \sum_{k,I_k<0} \frac{I_m I_k}{\sum_{t,I_t>0} I_t} \cdot min\{\frac{\sum_{t,I_t>0} I_t}{-\sum_{s,I_s<0} I_s}, 1\}$$
$$= r_m + I_m \cdot min\{1, \frac{-\sum_{k,I_k<0} I_k}{\sum_{t,I_t>0} I_t}\},$$

When $I_m < 0$,

$$\nu_m = max\{1 - \frac{\sum_{t,I_t>0} I_t}{-\sum_{s,I_s<0} I_s}(\frac{r_m - N_m U_m}{r_m}), \frac{N_m U_m}{r_m}\} r_m$$
$$= max\{r_m - \frac{\sum_{t,I_t>0} I_t}{-\sum_{s,I_s<0} I_s}(r_m - N_m U_m), U_m N_m\}$$
$$= max\{(1 - \frac{\sum_{t,I_t>0} I_t}{-\sum_{s,I_s<0} I_s})(r_m - N_m U_m) + U_m N_m, U_m N_m\}.$$

It is easy to verify that the chunk request distribution under this algorithm satisfies the optimal chunk request routing requirements (10) (**??**). Hence, the ISP-aware algorithm achieves the minimum chunk loss probability. For ISPs with $I_m < 0$,

$$L_m = max\{\frac{(1 - \frac{\sum_{t,I_t>0} I_t}{-\sum_{s,I_s<0} I_s})(r_m - N_m U_m)}{(1 - \frac{\sum_{t,I_t>0} I_t}{-\sum_{s,I_s<0} I_s})(r_m - N_m U_m) + U_m N_m}, 0\},$$

For ISPs with $I_m \geq 0$,

$$L_m = 0.$$

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

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

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 assume peers have the optimal cache distribution. This implies that we just need to consider the limitation of total peers' upload bandwidth.

We change the cross-ISP traffic through changing the proportion of chunk requests routed to other ISPs, $a'_{ml}s$. 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 by changing the value of a_{ml}'s. For ISPs with $I_m \geq 0$, let $a_{mm} = \epsilon_1, a_{ml} = (1-\epsilon_1) \cdot \frac{N_l}{N-N_m}, 1 \leq l \leq M, l \neq m$. For ISPs with $I_m < 0$, let $a_{mm} = \epsilon_2 \cdot max\{1 - \frac{\sum_{t,I_t>0} I_t}{-\sum_{s,I_s<0} I_s}(\frac{r_m - N_m U_m}{r_m}), \frac{N_m U_m}{r_m}\}$, $a_{ml} = min\{\frac{\sum_{t,I_t>0} I_t}{-\sum_{s,I_s<0} I_s}, 1\}(\frac{r_m - N_m U_m}{r_m}) \cdot \frac{I_l}{\sum_{t,I_t>0} I_t} + (1 - \epsilon_2) \cdot max\{1 - \frac{\sum_{t,I_t>0} I_t}{-\sum_{s,I_s<0} I_s}(\frac{r_m - N_m U_m}{r_m}), \frac{N_m U_m}{r_m}\} \cdot \frac{N_l}{N-N_m}$ for ISP l with $I_l \geq 0$, $a_{ml} = (1-\epsilon_2) \cdot max\{1 - \frac{\sum_{t,I_t>0} I_t}{-\sum_{s,I_s<0} I_s}(\frac{r_m - N_m U_m}{r_m}), \frac{N_m U_m}{r_m}\} \cdot \frac{N_l}{N-N_m}$ for ISP l with $I_l < 0$.

When $\epsilon_2 = \epsilon_1 \in (0, 1)$, the chunk requests routed to other ISPs increase. For ISPs with $I_m \geq 0$, we have

$$\nu_m = \sum_{l=1}^{M} a_{lm} r_l$$
$$= \epsilon_1 r_m + \sum_{k,I_k \geq 0} (1-\epsilon_1) \cdot \frac{N_m}{N - N_k} r_k +$$
$$min\{1, \frac{-\sum_{k,I_k<0} I_k}{\sum_{t,I_t>0} I_t}\} \cdot I_m + \sum_{k,I_k<0} (1-\epsilon_2)$$
$$\cdot max\{1 - \frac{\sum_{t,I_t>0} I_t}{-\sum_{s,I_s<0} I_s}(\frac{r_k - N_k U_k}{r_k}), \frac{N_k U_k}{r_k}\} \cdot \frac{N_m \cdot r_k}{N - N_k}$$

For ISPs with $I_m < 0$, we have

$$\nu_m = \sum_{l=1}^{M} a_{lm} r_l$$
$$= \epsilon_2 \cdot max\{1 - \frac{\sum_{t,I_t>0} I_t}{-\sum_{s,I_s<0} I_s}(\frac{r_m - N_m U_m}{r_m}), \frac{N_m U_m}{r_m}\} r_m$$
$$+ \sum_{k,I_k \geq 0} (1-\epsilon_1) \cdot \frac{N_m}{N - N_k} r_k + \sum_{k,I_k<0} [(1-\epsilon_2) \cdot max\{$$
$$1 - \frac{\sum_{t,I_t>0} I_t}{-\sum_{s,I_s<0} I_s}(\frac{r_k - N_k U_k}{r_k}), \frac{N_k U_k}{r_k}\} \cdot \frac{N_m}{N - N_k}] r_k$$

When $\epsilon_1 = 1$, $\epsilon_2 > 1$, the cross-ISP chunk requests are reduced. For ISPs with $I_m \geq 0$, $a_{mm} = 1$, $a_{ml} = 0$, for ISPs with $I_m < 0$, $a_{mm} = \epsilon_2 \cdot max\{1 - \frac{\sum_{t,I_t>0} I_t}{-\sum_{s,I_s<0} I_s}(\frac{r_m - N_m U_m}{r_m}), \frac{N_m U_m}{r_m}\}$, $a_{ml} = min\{\frac{\sum_{t,I_t>0} I_t}{-\sum_{s,I_s<0} I_s}, 1\}(\frac{r_m - N_m U_m}{r_m}) \cdot \frac{I_l}{\sum_{t,I_t>0} I_t} + (1 - \epsilon_2) \cdot max\{1 - \frac{\sum_{t,I_t>0} I_t}{-\sum_{s,I_s<0} I_s}(\frac{r_m - N_m U_m}{r_m}), \frac{N_m U_m}{r_m}\} \cdot \frac{N_l}{N-N_m}$ 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

$$\nu_m = \sum_{l=1}^{M} a_{lm} r_l$$
$$= r_m + min\{1, \frac{-\sum_{k,I_k<0} I_k}{\sum_{t,I_t>0} I_t}\} I_m + \sum_{k,I_k<0} (1-\epsilon_2)$$
$$\cdot max\{1 - \frac{\sum_{t,I_t>0} I_t}{-\sum_{s,I_s<0} I_s}(\frac{r_k - N_k U_k}{r_k}), \frac{N_k U_k}{r_k}\} \cdot \frac{N_m}{N - N_k} r_k$$

For ISPs with $I_m < 0$, we have

$$\nu_m = \epsilon_2 \cdot max\{1 - \frac{\sum_{t,I_t>0} I_t}{-\sum_{s,I_s<0} I_s}(\frac{r_m - N_m U_m}{r_m}), \frac{N_m U_m}{r_m}\} r_m$$

With ν_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:

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

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

V. Performance Evaluation

In this section, we carry out numerical analyses for relationship between chunk loss rates and cross-ISP traffic using parameters driven from the empirical data in the real-world [4]. There are in total $M = 10$ ISPs in the system. The total number of concurrent users in the system is $N = 100000$. The users distribute in different ISPs according to the probability distribution function $p_m = \frac{(M-m+1)^\beta}{\sum_{m=1}^{M}(M-m+1)^\beta}, \beta \geq 0$. $N_m = p_m \cdot N$. The user distribution among different ISPs is more unbalanced when β is larger. The average upload bandwidth of ISP m equals to $U_m = 1 + \frac{\gamma - m}{10}, 1 \leq \gamma \leq 10$. Chunk is the unit for storage and advertising to neighbors what parts of a movie a peer caches []. A chunk usually has a size of several MB. The total number of different chunks shared in the system is 5000. Every peer has a cache of 100 chunks. We use the Zipf-Mandelbrot model to fit the chunk popularity distribution: $\pi_j = \frac{\frac{1}{(j+q)^\alpha}}{\sum_{j=1}^{J}\frac{1}{(j+q)^\alpha}}$, $\alpha = 0.78, q = 4$. The number of peer neighbors is $d = 30$.

A. Optimal Cache vs. Unoptimal Cache

B. Cross-ISP Traffic vs. Chunk Loss Probability

We first evaluate different ISP's necessary cross-ISP traffic and chunk loss rate under ISP-aware chunk request routing algorithm. We simulate four scenarios:
Scenario 1: $\beta = 0.5, \gamma = 3$;
Scenario 2: $\beta = 1, \gamma = 3$;
Scenario 3: $\beta = 0.5, \gamma = 6$;
Scenario 4: $\beta = 1, \gamma = 6$;

Fig. 2 (a) & (b) plots each ISP's chunk loss rate and necessary cross-ISP traffic under ISP-aware chunk request routing algorithm. The cross-ISP traffic for ISP m is the traffic flowing into ISP m. They show that all ISPs reach 0 chunk loss rate when the total peers upload bandwidth in the system is enough to support video playback rate (Scenario 3 & Scenario 4). When the total peers upload bandwidth in the system is not enough to support video playback rate (Scenario 1 & Scenario 2), those ISPs with insufficient upload bandwidth suffer chunk loss. In all four scenarios, ISPs with insufficient upload bandwidth need to download contents from ISPs with sufficient upload bandwidth. The cross-ISP traffic is related to both peer number and peers' upload bandwidth, hence the traffic volume is not monotonically increasing.

Fig. 3 plots the change of the system chunk loss rate with the total cross-ISP traffic of all ISPs. The level of system chunk loss rate is mainly determined by peers' total upload bandwidth in the system. Scenario 1 & 2 have a much higher chunk loss rate than Scenario 3 & 4 as the peers' total upload bandwidth in Scenario 1 & 2 is not sufficient to support the playback rate. In this situation, the content providers should reduce the video streaming rate. In a specific scenario, the chunk loss rate is affected by the cross-ISP traffic. The VoD system achieves a minimum chunk loss rate under an appropriate cross-ISP traffic.

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

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

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

[4] J. Dai, B. Li, F. M. Liu, B. C. Li, and H. Jin, "On the Efficiency of Collaborative Caching in ISP-aware P2P Networks," in *Proc. of IEEE INFOCOM*, April 2011.

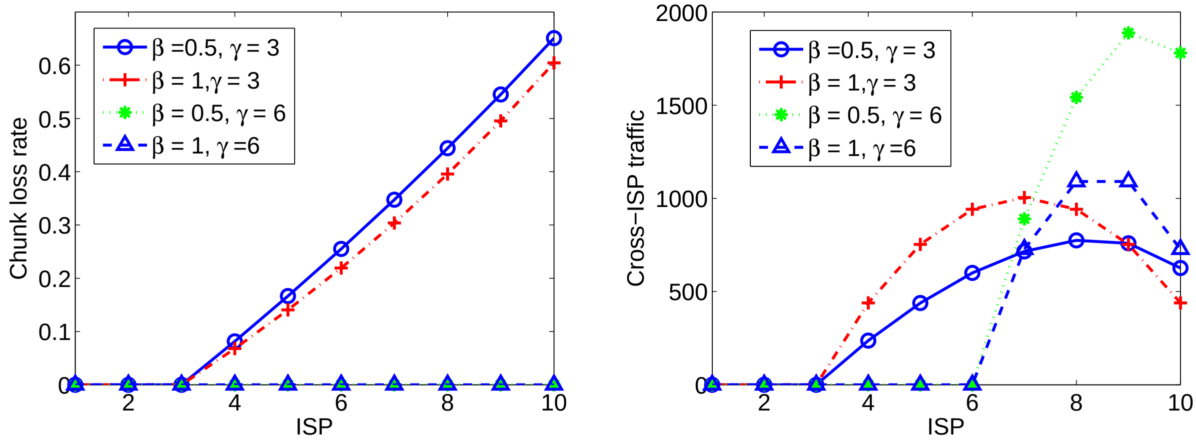

Figure 2. Chunk loss rate and cross-ISP traffic under ISP-aware chunk request routing algorithm.

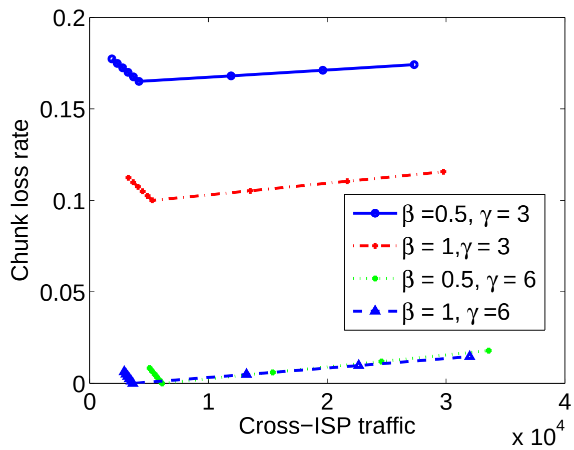

Figure 3. Relationship between chunk loss rate and cross-ISP traffic