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*— ISP-aware P2P applications have been proposed to reduce the significant amount of costly inter-ISP traffic. The impact of ISP-awareness on P2P system performance is not well understood. There is still lack of theoretical study on the relationship between performance and the volume of inter-ISP traffic. In this paper, we focus on the ISP-aware P2P 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. Our theoretical study not only help us understand the impact of ISP-awareness on the P2P VoD system performance, but also offer fundamental insights to designing ISP-aware P2P VoD systems.**

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. IV. We state the optimal peer cache placement strategy and the corresponding analytical chunk loss rates in Sec. V. We perform performance evaluation in Sec. VI, and conclude the paper in Sec. VII.

II. Model and Notation

We first introduce the VoD system model.

We consider a VoD system involving M ISPs. ISP m has totally N_m participating peers, with each peer either ON (*i.e.*, present in the system) or OFF (*i.e.*, logged off) at a time. The average peer upload bandwidth in ISP m is U_m. A 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 Θ be the set of all different cache states of peers, $W = |\Theta|$. The servers have cached all chunks.

A. Chunk Request

In the VoD system, different users may watch different parts or different channels of videos. Thus they request for downloading different video chunks. We model peers in different states according to the chunk they select to download. A peer that selects chunk c_j to download is at state $j, 1 \leq j \leq J$. The OFF peer is at state 0. The user behavior in the VoD system can be modeled as state transitions. It can be described by a transition matrix. In the matrix, element q_{jk} is the probability that a peer selects chunk c_k to download when finishing downloading chunk c_j. The stationary state distribution can be derived based on the transition matrix. $(\pi_0, \pi_1, \pi_2, ..., \pi_J)$ denote the stationary state distribution.

A request for a chunk is generated when a peer selects to download the chunk. One peer's number of requests for a specific chunk is a general renewal process with small intensity, as peers usually watch the same chunk after an enough long time. The requests for a chunk generated by peers in ISP m is the superposition of N_m peers' requests for the chunk. According to Palm-Khintchine theorem [7], the requests for a chunk generated by peers in ISP m can be modeled as a Poisson Process, with request rate $\lambda_{m,j} = N_m \cdot \pi_j$ for chunk c_j.

With the request rate for chunk j generated by peers in ISP m, $\lambda_{m,j}$, let us calculate the request rate for chunk j needed to be served by peers in ISP m, $\nu_{m,j}$. If a peer has already cached the chunk it is requesting, the request is served by local cache and no upload bandwidth is consumed. There is no need to consider this part of requests. The probability that a peer's request is not served by its local cache is Φ_j for chunk j. In reality, the case that a peer's request is served by the local cache usually corresponds that the user take a backward behavior to replay the video watched not long before. Let ϕ denote the probability that users take a backward behavior, $\Phi_j = 1 - \phi$. $\nu_{m,j}$ includes two parts, one is the request rate generated by ISP m and asking for service from peers in the same ISP; the other is the request rate generated by other ISPs and routed to ISP m. Let a_{mm} denote the fraction of requests asking for service from peers in the same ISP, ISP m. Let a_{lm} denote the fraction of requests routed from ISP l to ISP m. The request rate for chunk j needed to be served by peers in ISP m is:

$$\nu_{m,j} = \sum_{l=1}^{M} a_{lm} \cdot \Phi_j \cdot \lambda_{l,j}, 1 \leq j \leq J$$

The total chunk request rates generated by ISP m is $\lambda_m = \sum_{j=1}^{J} \lambda_{m,j} \cdot \Phi_j$

The total chunk request rates routed into ISP m is $R_m = \sum_{l=1}^{M} a_{lm} \cdot \lambda_l$

B. Peers' Cache Distribution

In the VoD system, a peer contributes some size of storage to cache chunks. The state of a peer's cache is $s_i = \{c_1^{(i)}, c_2^{(i)}, ..., c_B^{(i)}\}, 1 \leq i \leq W$. We do not consider the sequence of the cached contents, there are $W = C_J^B$ states. $N_m^{(i)}$ 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 $N_m \cdot \rho_j$.

C. ISP Awareness

The ISP-agnostic peer-to-peer (P2P) connections bring about large volume of inter-ISP traffic, among which most percent is unnecessary. To reduce the unnecessary inter-ISP traffic without deteriorating the VoD performance or even improving the performance, the ISP-aware P2P connections are adopted. In the ISP-aware P2P connections, peers keep track of which ISPs the connected neighbors are from. Peers adjust the number of internal neighbors and external neighbors according whether the intra-ISP resources are enough, route chunk requests to different ISPs according to the peers' resources in ISPs. Internal neighbors and external neighbors are selected from peers which have cached the chunk requested.

The proportion of chunk requests routed to external ISPs from ISP m is $1 - \frac{a_{mm}}{\sum_{l=1}^{M} a_{ml}}$. The proportion of chunk requests served by peers in the same ISP is $\frac{a_{mm}}{\sum_{l=1}^{M} a_{ml}}$.

D. Performance Metrics

The VoD streaming is a delay-sensitive application. The requests that can not be served by peers' resource are redirected to servers. When server rate is not enough to support the requests, some requests are dropped, which results in loss rate for chunks.

$P_{m,j}$ denotes the acceptance probability for the requests for chunk j asking for service from ISP m. P_m denotes the average acceptance probability for the requests asking for service from ISP m. We use the average chunk loss rate under no server capacity used to represent the system performance. The average chunk loss rate for ISP m, L_m is:

$$L_m = 1 - \sum_{l=1}^{M} a_{ml} \cdot P_l$$

$$P_l = \frac{\sum_{j=1}^{J} P_{l,j} \cdot \nu_{m,j}}{\sum_{j=1}^{J} \nu_{m,j}}$$

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

III. Loss network model for VoD

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 present the corresponding results of [9] to our present purpose.

$\mathbf{n_m} = \{n_{m,j}\}_{c_j \in \mathcal{C}}$ denotes the vector of request numbers for different chunks being served concurrently in ISP 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.
$\lambda_{m,j}$	the request rate for chunk j generated by peers in ISP m.
$\nu_{m,j}$	the request rate for chunk j asking for service from ISP m.
a_{lm}	the fraction of requests routed from ISP l to ISP m.
Φ_j	the probability that a peer's requests for chunk j is not served by itself cache.
$L_{m,j}$	the loss rate for chunk j in ISP m.
$y_{out,m}$	the number of external neighbors at a peer in ISP m.
$y_{in,m}$	the number of internal neighbors at a peer in ISP m.
d	number of active neighbors at a peer.

$L_{m,j}$ denotes the fraction of dropped request numbers for chunk j in ISP m. The requests under service experience a delay of one unit time (service time). The loss requests experience a delay of 0. The average delay experienced by chunk requests, $D_{m,j}$, 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

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

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:

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

The corresponding Lagrangian is:

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

The KKT conditions for this convex optimization problem

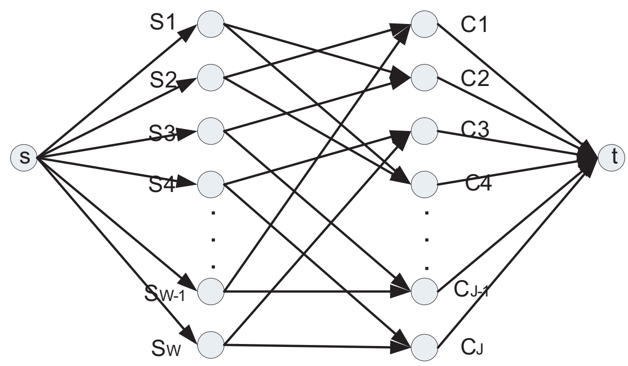

Figure 1. Corresponding Bipartite Graph.

are:

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

$$\epsilon_{\mathcal{A}} \ge 0 \tag{2}$$

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

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

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

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

IV. MAXIMUM-BIPARTITE-FLOW MAP OF THE KKT SOLUTIONS

The served request rates are 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.

We first give 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 Θ and $\mathcal{C}$ with edges directed from Θ to $\mathcal{C}$. The left set of nodes, Θ, 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 Θ, 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 Θ 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 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.*

Proof: We first prove that the served request rates obtained by solving the KKT conditions are the flow from nodes in set $\mathcal{C}$ to destination t under the maximum bipartite flow.

$x_{m,j}$ are the solutions of the KKT conditions for the served request numbers. We can divide the $x_{m,j}$ into two classes according whether $x_{m,j}$ is equal to $\nu_{m,j}$. $\mathcal{C}_1 = \{c_j | x_{m,j} = \nu_{m,j}\}$, $\mathcal{C}_2 = \{c_j | x_{m,j} < \nu_{m,j}\}$. From the KKT conditions, we can get $\forall \mathcal{A}$, when $\mathcal{A}$ includes $c_j \in \mathcal{C}_1$, $\epsilon_{\mathcal{A}} = 0$, when the elements in $\mathcal{A}$ are all from $\mathcal{C}_2$, $\epsilon_{\mathcal{A}} \neq 0$. The peers' upload bandwidth for the chunks in set $\mathcal{C}_2$ is not enough. According to KKT condition (3), $\forall \mathcal{A} \subseteq \mathcal{C}_2$, $\sum_{c_j \in \mathcal{A}} x_{m,j} = U_m \cdot \sum_{i: s_i \cap \mathcal{A} \neq \emptyset} N_m^{(i)}$. Next, we just need to prove that the cutset from the set including destination node t, 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 the minimum cut.

First, when the set including destination node t contains a node c_j from $\mathcal{C}_1$, as the capacity of edges from nodes in Θ to node c_j is equal to that of the edge from source s to nodes in Θ. So, to minimize the change of capacity of cutset, all nodes in Θ having connections with c_j should be included in the set. As a result, the total capacity of cutset is increased by $U_m \cdot \sum_{s_i : c_j \in s_i} N_m^{(i)} - nu_{m,j}, c_j \in \mathcal{C}_1$. When the set including destination node t excludes a node c_k from $\mathcal{C}_2$, to minimize the change of capacity of cutset, the nodes in Θ having connections with c_k should also be excluded. At last, the total capacity of cutset is increased by $\nu_{m,j} - U_m \cdot \sum_{s_i : c_k \in s_i} N_m^{(i)}, c_k \in \mathcal{C}_2$.

A. Maximum Bipartite Flow

Based on Theorem 1 and Equation (5), we can get the average chunk loss rate in different ISPs through solving the maximum bipartite flow. We apply the modified push-relabel algorithm (Algorithm 1) to obtain the maximum bipartite flow. In the VoD system, one peer caching B different chunks can share its upload bandwidth between those requests for the B chunks. Peers at state s_i push their flow proportionally to the chunk request rates.

We can obtain the total served request rates through the algorithm, which is $\sum_{j=1}^{J} flowm(j)$. According to Theorem 1, $\sum_{j=1}^{J} x_{m,j} = \sum_{j=1}^{J} flowm(j)$.

The algorithm divides the nodes in set Θ into three categories with different heights, $height = 0$, $height = 2$, $height = height(s) + 1$. Nodes with $height = 0$ mean no peers are at these cache states. Nodes with $height = 2$ mean peers at these cache states are just enough or not sufficient. Nodes with $height = height(s) + 1$ mean more than enough peers are at these cache states.

The algorithm divides the nodes in set $\mathcal{C}$ into two categories with different heights, $height = 1$, $height = 3$. Nodes with $height = 3$ in set $\mathcal{C}$ mean the request numbers for these chunks can be satisfied. Nodes with $height = 1$ in set $\mathcal{C}$ may have a loss rate.

Algorithm 1: Maximum Bipartite Flow

Input: the total upload bandwidth of peers at state i, $N_m^{(i)} U_m$, $1 \leq i \leq W$;
the request rate for chunk j, $\nu_{m,j}$, $1 \leq j \leq J$.
Output: the served request rate for chunk j, $flowm(j)$, $1 \leq j \leq J$.

Initialize the height, excess of each vertex as 0, except $height(s) = |V|$;
Initialize the flow of each edge as 0;
Source s pushes flow to each vertex $s_i \in \Theta$, update the flow between s and each vertex $s_i \in \Theta$, the excess of $s_i \in \Theta$;
Increase the height of vertices in Θ with $excess \neq 0$ to 2;
Increase the height of all vertices in $\mathcal{C}$ to 1;
Set P as the set of vertices in Θ with $height = 2$ and $excess \neq 0$;
while *P is not empty.* **do**
- **for** *each vertex $s_i \in P$* **do**
  - Calculate the total request rate of the chunks state s_i caches, $R(s_i)$;
  - s_i pushes flow : $d(s_i, c_j) = excess(s_i) \cdot \frac{\nu_{m,j}}{R(s_i)}$ to c_j, the chunks it caches with $height(c_j) = 1$;
  - Update the excess of vertices s_i, c_j, the flow between s_i and c_j, the set $P = P - s_i$;
- **for** *each vertex $c_j \in \mathcal{C}$ with $excess \neq 0$ and $height = 1$* **do**
  - c_j pushes flow to t: $d(c_j, t) = min(excess(c_j), \nu_{m,j} - flowm(j))$;
  - Update the excess of vertices $c_j \in \mathcal{C}$ and the flow between c_j and t: $flowm(j) = flowm(j) + d(c_j, t)$;
  - **if** $excess(c_j) \neq 0$ **then**
    - $height(c_j) = 3$;
- **for** *each vertex $c_j \in \mathcal{C}$ with $excess \neq 0$ and $height = 3$* **do**
  - Divide vertices $s_i \in \Theta$ caching c_j into two groups: H, for $s_i \in H$, all vertices that vertex s_i connects to have height 3; L, for $s_i \in L$, existing vertices that vertex s_i connects to have height 1;
  - **if** *L is not empty.* **then**
    - c_j pushes flow back to vertices in L: $d(c_j, s_i) = min(f(s_i, c_j), excess(c_j) \cdot \frac{\frac{1}{R(s_i)}}{sum_{s_i : s_i \in L} \frac{1}{R(s_i)}})$;
    - Update the excess and flow and set $P = P + s_i$;
  - **if** *L is empty.* **then**
    - c_j pushes flow back to nodes in H: $d(c_j, s_i) = excess(c_j) \cdot \frac{\frac{1}{R(s_i)}}{sum_{s_i : s_i \in L} \frac{1}{R(s_i)}})$.
    - Update the excess, flow, height: $height(s_i) = height(s) + 1$;

V. PERFORMANCE ANALYSIS UNDER OPTIMAL CACHE

The maximum bipartite flow algorithm can solve the average request acceptance rate under a specific chunk request rate distribution and a specific cache distribution. Through the algorithm, we can also obtain some insights on what's the optimal cache in ISP-aware VoD, and analytically derive the corresponding request acceptance rate.

A. Optimal Cache in ISP-aware VoD

Through the maximum bipartite flow algorithm, we conclude that the cache placement strategy in which no cache states become $height = height(s) + 1$ after the algorithm is an optimal cache placement strategy. As peers can adjust their upload bandwidth share for different cached chunks, the optimal cache placement strategy is not unique.

Lemma 1. *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* $\gamma_i = \frac{\sum_{c_j \in s_i} \nu_{m,j}}{C_{J-1}^{B-1} \sum_{j=1}^{J} \nu_{m,j}}$, *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* $\rho_j = \frac{\nu_{m,j}}{\sum_{j=1}^{J} \nu_{m,j}}$. *Peers allocating their upload bandwidth uniformly among cached chunks can achieve the optimal performance.*

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

B. Analysis of Least Recently Used Cache Replacement Strategy

In this section, we theoretically prove that the LRU cache replacement strategy achieve the optimal cache in the stationary states.

We analyze the stationary probability that a peer caches chunk j. There are total B cells in a peer's cache. Cell 1 caches the most recently played chunks. Cell B caches the chunk played the longest before. Let $P_j(b)$ denote the probability that cell b caches chunk j. In the stationary states, the probability $P_j(b)$ equals to the sum of two parts, one is the probability that the cell $b-1$ caches chunk j and is moved to cell b; the second is the probability that cell b caches chunk j and stay unmoved. For $2 \le b \le B$,

$$\begin{aligned} P_j(b) &= (1-\phi)P_j(b-1) + \\ &\phi \cdot \frac{B-b+1}{B} P_j(b-1) + \phi \cdot \frac{b-1}{B} P_j(b) \\ P_j(b)(1-\phi\frac{b-1}{B}) &= P_j(b-1)(1-\phi\frac{b-1}{B}) \\ P_j(b) &= P_j(b-1) \end{aligned}$$

Thus, $P_j(1) = P_j(2) = \ldots = P_j(B)$.

$$P_j(1) = (1-\phi)\pi_j + \phi[\sum_{b=1}^{B} \frac{1}{B} P_j(b)]$$

$$P_j(1) = \pi_j$$

So, the probability that a peer cache chunk j is π_j under the stationary states of LRU replacement strategy.

C. System Performance under Optimal Cache

The average request acceptance probability in ISP m is $P_m = min\{\frac{U_m \cdot N_m}{\sum_{j=1}^{J} \nu_{m,j}}, 1\}$ under optimal cache.

The average chunk loss rate for the system is:

$$\begin{aligned} L &= \frac{\sum_{m=1}^{M}(1-P_m)\cdot R_m}{\sum_{m=1}^{M} R_m} \\ &= \frac{\sum_{m=1}^{M} R_m - \sum_{m=1}^{M} R_m P_m}{\sum_{m=1}^{M} R_m} \\ &= 1 - \frac{\sum_{m=1}^{M} R_m P_m}{\sum_{m=1}^{M} R_m} \\ &\ge 1 - \frac{\sum_{m=1}^{M} U_m N_m}{\sum_{m=1}^{M} R_m} \\ &\ge 1 - \frac{\sum_{m=1}^{M} U_m N_m}{(1-\phi)N} \end{aligned}$$

In reality, the system demands the chunk loss rate be 0, $L = 0$, or the system has some resource redundancy, $L < 0$. $1 - \frac{\sum_{m=1}^{M} U_m N_m}{(1-\phi)N} \le 0$, when no backwards are taken by users, $r \le \frac{\sum_{m=1}^{M} U_m N_m}{N}$. The user backwards behaviors make the system chunk loss rate reach lower.

D. Number of Inter-ISP Connections

For the homogeneous case:

When the total peer upload bandwidth in ISP m, $N_m U_m > N_m r$, peers in other ISPs can connect to peers in ISP m requesting for chunks. Under the LRU cache replacement strategy, the corresponding peer upload bandwidth allocation strategy is uniform allocation among different chunks. Thus, the average upload bandwidth of one connection is U_m/B. The number of connections transmitting contents out of ISP m is $\frac{N_m U_m - N_m r}{U_m/B} = N_m B(1 - \frac{r}{U_m})$.

When the total peer upload bandwidth in ISP m, $N_m U_m < N_m r$, peers in ISP m connect to peers in other ISPs to request for chunks. The number of connections transmitting chunks from other ISPs to ISP m is $\frac{N_m r - N_m U_m}{U_m/B} = N_m B(\frac{r}{U_m} - 1)$.

For the heterogeneous case:

Assume there are mainly two kinds of peers' upload bandwidth, one is smaller than the playback rate, the other is larger than the playback rate. Due to the dynamic log on and off of peers, we can calculate the probability that there are at least k inter-ISP connections needed for an ISP.

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.

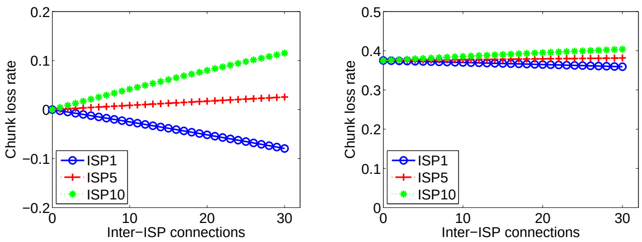

Figure 2. Chunk loss rate and inter-ISP connections relationship under optimal and unoptimal peer cache distribution.

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.