Stochastic Model for ISP-aware VoD Streaming

***Abstract*— The abstract goes here.**

I. Introduction

The proliferation of high-speed broadband services make people start to enjoy watching online videos. 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, UUSee*, 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 the motivation to consider 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-friendly P2P applications are proposed.

II. Motivation

The peer-to-peer (P2P) technology has been popularly adopted to deliver contents in the Internet. As P2P technology exploits distributed peers' resources to alleviate servers' workload, its ISP-agnostic property brings about large volume of cross-ISP Internet traffic, which increases the cost of ISPs. Several ISPs have started to limit the bandwidth consumption of P2P applications by proactively detecting and throttling their data packets. To become ISP-aware and avoid the blocking of ISPs, peers try to control the cross-ISP connections to reduce the unnecessary cross-ISP traffic. In this paper, we analyze the effect of controls of cross-ISP connections on Video-on-Demand (VoD) system performance. In ISP-aware VoD systems, peers record the number of internal and external neighbors. Based on the analysis, we propose under which cases and by how much should the restrictions of cross-ISP connections be. We also analyze the best peer cache distribution for the ISP-aware VoD.

III. Model and Notation

We first introduce the VoD system model.

We consider a VoD system involving $M = |\mathcal{M}|$ ISPs. ISP i has totally $N_i = |\mathcal{P}_i|$ 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 i is U_i. A VoD system supplies multiple video channels. As a peer can watch any chunks in any video at a time, we do not focus on the notion of videos or channels, instead focus on a collection of $J = |\mathcal{C}|$ chunks, $\mathcal{C} = \{c_1, c_2, ..., c_J\}$, regardless of which video they belong to. The playback time for one chunk is one unit time. A peer $p \in \mathcal{P}_i, i \in \mathcal{M}$ can cache and serve chunks from different videos. Every peer has a cache size B. Denote $\mathcal{B} = \{c^{(1)}, c^{(2)}, ..., c^{(B)}\}$ as the cache state of a peer. We model the servers as special peers that have cached all chunks. Denote the server capacity in ISP i as S_i.

A. Chunk Request

In the VoD system, 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. Denote $(\pi_0, \pi_1, \pi_2, ..., \pi_J)$ as the stationary state distribution. In stationary states, $m_{i,j} = N_i \cdot \pi_j$ peers are at state j.

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. The requests for a chunk generated by peers in ISP i is the superposition of N_i peers' requests for the chunk. According to Palm-Khintchine theorem, the requests for a chunk generated by peers in ISP i can be modeled as a Poisson Process, with request rate $\lambda_{i,j} = N_i \cdot \pi_j$ for chunk c_j.

With the request rate for chunk j generated by peers in ISP i is $\lambda_{i,j}$, let us calculate the request rate for chunk j needed to be served by peers in ISP i, $\nu_{i,j}$. If a peer has already cached the chunk it requested, 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. $\nu_{i,j}$ includes two parts, one is the request rate generated by ISP i and asking for service from peers in the same ISP; the other is the request rate generated by other ISPs and routed to ISP i. Denote a_{ii} as the fraction of requests asking for service from peers in the same ISP, ISP i. Denote a_{mi} as the fraction of requests routed from ISP m to ISP i. The request rate for chunk j in ISP i is:

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

B. Chunk Distribution in Peers' Cache

In the VoD system, a peer contributes some size of storage to cache chunks. The state of a peer's cache is $\mathcal{B} =$

$\{c^{(1)}, c^{(2)}, ..., c^{(B)}\}$. If not consider the sequence of the cached contents, there are $W = C_J^B$ states. If we consider the sequence of the cached contents, there are $W = B! \cdot C_J^B$ states. $\mathcal{W}$ denotes the set of different cache states. The stationary distribution of cache states under a specific cache placement strategy is $\gamma_{\mathcal{B}}$, for state$\mathcal{B}$. A simple placement strategy is Least Recently Used (LRU). In VoD systems adopting LRU, at each time slot, a peer's cache is kept as a sequence of chunks arranged in order of latest access time. $(c_t^{(1)}, c_t^{(2)}, ..., c_t^{(B)})$ denote the state of a peer's cache at time slot t. The LRU caching model is an irreducible and aperiodic Markov Chain. The stationary probability of cache states exists.

C. Performance Metrics

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

$L_{i,j}$ denotes the loss rate for requests of chunk j in ISP i when no server capacity is used. The total needed server capacity in ISP i to make 0 loss rate is:

$$S_i = \sum_{j=1}^{J} L_{i,j} \cdot \nu_{i,j}$$

D. 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 that whether the intra-ISP resources are enough. Internal neighbors and external neighbors are selected uniformly. $x_{out,i}$ denotes the number of external neighbors of peers in ISP i. d_{in} denotes the number of neighbors. The number of internal neighbors is:

$$x_{in,i} = d_{in} - x_{out,i}.$$

A peer sends its requests for chunks uniformly to its neighbors. The fraction of requests for chunk j generated by peers in ISP i and not routed to other ISPs is $a_{ii} = \frac{x_{in,i}}{d_{in}}$. The fraction of requests for chunk j from ISP m to ISP i is $a_{mi} = \frac{N_i}{N-N_m} \frac{x_{out,m}}{d_{in}}$. The total request rate for chunk j in ISP i is:

$$\nu_{i,j} = \frac{x_{in,i}}{d_{in}} \cdot \Phi_j \cdot \lambda_{i,j} + \sum_{m=1,m\neq i}^{M} \frac{N_i}{N-N_m} \frac{x_{out,m}}{d_{in}} \cdot \Phi_j \cdot \lambda_{m,j}$$

As the served requests will induce traffic, the cross-ISP traffic from other ISPs to ISP i is:

$$T_{in,i} = \sum_{j=1}^{J} \sum_{m=1,m\neq i}^{M} \cdot \frac{N_m}{N-N_i} \frac{x_{out,i}}{d_{in}} \Phi_j \lambda_{i,j}$$

TABLE I.
IMPORTANT NOTATIONS

M	number of ISPs
N	total number of peers in the system
N_i	number of peers in ISP i
U_i	average peer upload bandwidth in ISP i
S_i	actual server capacity deployed in ISP i
B	the cache size of a peer
J	the number of chunks shared in VoD
$\lambda_{i,j}$	the request rate for chunk j generated by peers in ISP i
$\nu_{i,j}$	the request rate for chunk j asking for service from ISP i.
a_{mi}	the fraction of requests routed from ISP m to ISP i.
Φ_j	the probability that a peer's requests for chunk j is not served by itself cache.
$L_{i,j}$	the loss rate for chunk j in ISP i.
$x_{out,i}$	the number of external neighbors at a peer in ISP i
$x_{in,i}$	the number of internal neighbors at a peer in ISP i
d_{in}	number of active neighbors at a peer
$T_{out,i}$	the cross-ISP traffic flowing out of ISP i
$T_{in,i}$	the cross-ISP traffic flowing into ISP i from other ISPs

The cross-ISP traffic from ISP i to other ISPs is:

$$T_{out,i} = \sum_{j=1}^{J} \sum_{m=1,m\neq i}^{M} \frac{N_i}{N-N_m} \frac{x_{out,m}}{d_{in}} \cdot \Phi_j \cdot \lambda_{m,j}$$

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

IV. Loss Network Analysis

The chunk requests in the P2P assisted VoD system are first submitted to peers in the P2P system. If they are accepted, peers' upload bandwidth is used to serve them at the video streaming rate. If no enough resources from peers can support them, they are rejected and redirected to servers. This model ensures zero waiting time for requests, which is desirable for VoD application. We first consider the system with no servers to analyze the rejected request rates. The needed server capacity can be derived from the rejected request rates. The system can be analyzed using loss network model. 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.

$\mathbf{n_i} = \{n_{i,j}\}_{c_j \in \mathcal{C}}$ denotes the vector of request numbers for different chunks being served concurrently in ISP i. The vector $\mathbf{n_i}$ of request numbers under service is a particular instance of the loss network model. $\mathbf{n_i}$ has a product-form stationary distribution:

$$\pi(\mathbf{n_i}) = \frac{1}{G} \prod_{c_j \in C} \frac{\nu_{i,j}^{n_{i,j}}}{n_{i,j}!}$$

where G is the normalizing constant.

The state space of the vector $\mathbf{n_i}$ satisfies some constraints.

d_{jp} denotes the number of concurrent downloads of chunk c_j from peer p. The following two constraints should be satisfied in ISP i:

$$\sum_{p:p\in\mathcal{P}_i, c_j\in\mathcal{B}_p} d_{jp} = n_{i,j}, \forall c_j \in \mathcal{C}$$
$$\sum_{j:c_j\in\mathcal{B}_p} d_{jp} \leq U_i, \forall p \in \mathcal{P}_i$$

$\mathcal{B}_p$ is the cache state of peer p.
A compact characterization of constraints is:

$$\forall \mathcal{A} \subseteq \mathcal{C}, \sum_{c_j\in\mathcal{A}} n_{i,j} \leq U_i \cdot |p \in \mathcal{P}_i : \mathcal{A} \cap \mathcal{B}_p \neq \emptyset|$$

Which can be written as:

$$\forall \mathcal{A} \subseteq \mathcal{C}, \sum_{c_j\in\mathcal{A}} n_{i,j} \leq U_i \cdot \sum_{\mathcal{B}:\mathcal{B}\cap\mathcal{A}\neq\emptyset} N_{\mathcal{B},i}$$

$N_{\mathcal{B},i}$ is the number of peers at cache state $\mathcal{B}$ in ISP i.

$L_{i,j}$ denotes the fraction of dropped request numbers for chunk j in ISP i. 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 is:

$$D_{i,j} = (1 - L_{i,j}) \cdot 1 + L_{i,j} \cdot 0 = (1 - L_{i,j})$$

Upon applying Little's law to the system (with respect to chunk j), we obtain

$$\nu_{i,j} D_{i,j} = \mathbf{E}[n_{i,j}]$$

which yields

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

Due to the computational complexity of the exact stationary distribution, we take the 1-point approximate algorithm for computing the stationary loss probabilities. In the 1-point approximation, use $n^*_{i,j}$, which is the element of $\mathbf{n_i^*}$, the state having the maximum probability as a surrogate of $\mathbf{E}[n_{i,j}]$. $\mathbf{n_i^*}$ is the solution of the optimization problem:

$$\text{maximize} \sum_{j=1}^{J} n_{i,j} log\nu_{i,j} - logn_{i,j}!$$
$$\text{over} \forall \mathcal{A} \subseteq \mathcal{C}, \sum_{c_j\in\mathcal{A}} n_{i,j} \leq U_i \cdot |p \in \mathcal{P}_i : \mathcal{A} \cap \mathcal{B}_p \neq \emptyset|$$
$$\mathbf{n_i} \geq 0$$

We replace $logn_{i,j}!$ by $n_{i,j}logn_{i,j} - n_{i,j}$ according to Stirling's formula, $logn_{i,j}! = n_{i,j}logn_{i,j} - n_{i,j} + O(logn_{i,j})$. Relax integer vector $\mathbf{n_i}$ using a real vector $\mathbf{x_i}$.

$$\text{maximize} \sum_{j=1}^{J} x_{i,j} log\nu_{i,j} - x_{i,j}logx_{i,j} + x_{i,j}$$
$$\text{over} \forall \mathcal{A} \subseteq \mathcal{C}, \sum_{c_j\in\mathcal{A}} x_{i,j} \leq U_i \cdot |p \in \mathcal{P}_i : \mathcal{A} \cap \mathcal{B}_p \neq \emptyset|$$
$$\mathbf{x_i} \geq 0$$

The corresponding Lagrangian is:

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

The KKT conditions for this convex optimization problem are:

$$\forall \mathcal{A} \subseteq \mathcal{C}, \sum_{c_j\in\mathcal{A}} x_{i,j} \leq U_i \cdot \sum_{\mathcal{B}:\mathcal{B}\cap\mathcal{A}\neq\emptyset} N_{\mathcal{B},i}$$
$$\epsilon_{\mathcal{A}} \geq 0$$
$$\forall \mathcal{A} \subseteq \mathcal{C}, \epsilon_{\mathcal{A}} \cdot (U_i \cdot \sum_{\mathcal{B}:\mathcal{B}\cap\mathcal{A}\neq\emptyset} N_{\mathcal{B},i} - \sum_{c_j\in\mathcal{A}} x_{i,j}) = 0$$
$$x_{i,j} = \nu_{i,j} \cdot \exp(-\sum_{c_j\in\mathcal{A},\mathcal{A}\subseteq\mathcal{C}} \epsilon_{\mathcal{A}})$$

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

The optimal value for the optimization problem can be got by solving the KKT conditions. The number of functions in KKT conditions grows exponentially with the number of chunks. We prove that the solution of the KKT conditions can be mapped into the maximum bipartite flow.

We first give the corresponding bipartite graph (Fig. **??**): along with the source node s and the destination node t, the bipartite graph has two sets of nodes $\mathcal{C}$ and $\mathcal{W}$ with edges directed from $\mathcal{C}$ to $\mathcal{W}$. The left set of nodes, $\mathcal{C}$, represents different chunks, the right set of nodes, $\mathcal{W}$, represents peers with different cache states. The edges directed from node j, representing chunk j in $\mathcal{C}$ to nodes having cached chunk j in $\mathcal{W}$ represent the flow of requests for chunks. The edges from source s to any node in set $\mathcal{C}$ represent the requests for chunk c_j, and have the capacity, $\nu_{i,j}$. The edges from $c_j \in \mathcal{C}$ to $\mathcal{B} \in \mathcal{W}, c_j \in \mathcal{B}$ have the capacity, $\nu_{i,j}$, which can not exceed the total request rate for chunk j. The edges from any nodes in $\mathcal{W}$ to the destination t represent the requests served by peers, having a capacity of $N_{\mathcal{B}} \cdot U_i$.

Theorem 1. *The served request numbers obtained from the maximum bipartite flow and those obtained from the KKT conditions are the same.*

Proof:
i) We first prove that the served request numbers obtained by solving the KKT conditions are the flow from source to different chunks under the maximum bipartite flow.

$x_{i,j}$ are the solutions of the KKT conditions for the served request numbers. We can divide the $x_{i,j}$ into two classes according whether $x_{i,j}$ is equal to $\nu_{i,j}$. $\mathcal{C}_1 = c_j | xi, j = \nu_{i,j}$, $\mathcal{C}_2 = c_j | xi, j < \nu_{i,j}$. 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 $\mathcal{C}_2$, $\epsilon_{\mathcal{A}} \neq 0$. So, for all $c_j \in \mathcal{C}_2$, $\sum_{c_j \in \mathcal{C}_2} x_{i,j} = U_i \cdot \sum_{\mathcal{B}:\mathcal{B}\cap\mathcal{A}\neq\emptyset} N_{\mathcal{B}}$. Next, we just need to prove that the cutset from the set including source node s, all nodes in set $\mathcal{C}_2$, nodes in set $\mathcal{W}$ 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 source node s contains a node c_j from $\mathcal{C}_1$, as the capacity of edges from node c_j to nodes in $\mathcal{W}$ is equal to that of the edge from source s to node c_j in $\mathcal{C}_1$. So, to decrease the capacity of cutset, all nodes having connections with c_j in $\mathcal{W}$ should be included in the set. The capacity of cutset is increased by $U_i \cdot \sum_{\mathcal{B}:c_j\in\mathcal{B}} N_{\mathcal{B}} - x_{i,j}$. When the set including source node s excludes a node c_k from $\mathcal{C}_2$, to decrease the capacity of cutset, the nodes in $\mathcal{W}$ having connections with c_k should also be excluded. The capacity of cutset is increased by $\nu_{i,j} - U_i \cdot \sum_{\mathcal{B}:c_j\in\mathcal{B}} N_{\mathcal{B}}$.

ii) We then prove that the served request numbers obtained from maximum bipartite flow problem satisfies the KKT conditions.

1)Set $y_{i,j}$ as the request rate for chunk c_j that are the results of the solution of maximum flow problem.
2)Using max-flow min-cut theorem, show that $y_{i,j}$ can be expressed as the form of $y_{i,j} = \nu_{i,j} \cdot \prod_{c_j\in\mathcal{A},\mathcal{A}\subseteq\mathcal{C}} \eta_{\mathcal{A}}, \eta_{\mathcal{A}} = exp(-\epsilon_{\mathcal{A}})$.
3) It is obvious that $\forall \mathcal{A} \subseteq \mathcal{C}, \sum_{c_j\in\mathcal{A}} y_{i,j} \leq N_i \cdot U_i \cdot \sum_{\mathcal{B}:\mathcal{B}\cap\mathcal{A}\neq\emptyset} \gamma_{\mathcal{B}}$
4) Prove that $\forall \mathcal{A} \subseteq \mathcal{C}, N_i \cdot U_i \cdot \sum_{\mathcal{B}:\mathcal{B}\cap\mathcal{A}\neq\emptyset} \gamma_{\mathcal{B}} - \sum_{c_j\in\mathcal{A}} y_{i,j} = 0, \text{or} \eta_{\mathcal{A}} = 1$.

A. Problems to be analyzed

1. We restrict a peer's cross-ISP connections, which is ISP-aware. We calculate the chunk request rate in one ISP under the restriction.

2. apply the max-flow min-cut theorem to analyze the performance of a fixed cache distribution; under a specific chunk request rate distribution;

3. What conditions should the best cache distribution satisfy? How is it determined by the chunk request rate?

4. Does a simple cache strategy exist?

VI. Conclusions