Modeling and Performance analysis of HnH: a new approach of combining Live Streaming Channels having a small number of participants

I. Shahriar, D. Qiu, B. Jaumard

September 21, 2011

Abstract—In this work, our focus is on the Dedicated Channels having a Small-numbered Viewers (DCSV for short) in a multi-channel live streaming system. Usually, these are dedicated and user generated channels and suffer adversely from poor channel performance, mainly, due to having a small number of participants. As a result, when a viewer explicitly requests for a block of streaming content, the probability that the block of data will be available among the existing viewers is less than it would be if the number of viewers was higher in that channel (e.g., viewers in a popular channel). We propose a new approach of cross-channel resource sharing scheme, named it as HnH (short for Hand in Hand) in order to solve the performance problem of those DCSV channels. In this work, we propose a discrete-time stochastic model to analyze the performance issues of the proposed HnH scheme and provide insight into it.

Index Terms—P2P, Live streaming, Multi-channel systems, Stochastic model, Streaming buffer, Playback pointer.

I. INTRODUCTION

In recent years, Internet has witnessed a rapid growth in P2P applications, especially, in the live streaming domain. There have been several deployments of large-scale industrial P2P live video systems, e.g., CoolStream [1], PPLive [2], Sopcast [3]. Several contemporary measurement studies have verified that thousands of users can simultaneously participate in these systems. Almost all live P2P video systems offer multiple channels (e.g., PPLive [2] can host over 100 channels). It is expected that in near future, live streaming systems with hundreds of user-generated channels and dedicated channels will likely have thousands of live channels in total. However, it has been found in the literature (e.g., [4]) that these user-generated channels and dedicated channel are having poor performance, mainly, due to smaller number of participants. In order to solve this issue, we proposed a new cross-channel resource cooperation scheme (in another work) called HnH. However, we do not cover the details of the HnH scheme here.

In short, in the HnH scheme, peers from two or more DCSV channels will cooperate each other for the common goal of having better performance. Thus, they form a combined HnH channel and increase the number of participants for each of the channels in cooperation, which was one of the main issues for the performance problem in DCSV channels.

In this work, we assume that a HnH system is already in place and we develop stochastic models for the different scenarios of a multi-channel HnH live streaming system.

II. RELATED WORKS

A. File sharing systems

Qiu et al. [5] have proposed a stochastic model of for a Bittorrent-like P2P file sharing system. Their work is motivated by the fact that the upload BW utilization of a peer is depends on how many pieces that peer has in its download buffer. In this model they shown the peer distribution $P_i$, i.e., the probability that a randomly selected peer has $i$ useful pieces of file content where $0 \leq i \leq N$ and $N$ is the actual (total) number of pieces of the entire file. The number of neighbors is assumed to be a fixed number for all the peers. They have assumed that the peer distribution does not change with time once a peer reaches the the steady-state phase. Thus the peer distribution is calculated from the average download rate of a peer with $i$ pieces. Later on they have considered the seed departure rate, peer arrival rate and total number of peers present in a system for a given time while the system is in steady state. They have provided numerical solution. They have provided numerical results for i) Piece distribution ii) Download rate distribution and iii) performance of the system due to seed departure, based on number of pieces and number of neighbors.

Our work is motivated by the work of Qiu et al. [5]. However, modeling a P2P live streaming system differs from modeling a file sharing system in many aspects due to different nature of them. A few key issues to be mentioned are: (i) small playback buffer (ii) pieces are played back in real time, late arrival or too early arrival of pieces do not help the system (iii) even if a neighbor may have many useful pieces, a peer may not request for those pieces due to relative position of their playback pointers, etc.

B. Live streaming systems

Issaei et al. [6] proposed a stochastic model for the Efficiency of BitTorrent-like Peer-to-Peer Live Streaming. They have studied the effects of various network parameters for the streaming system having peers with a single channel and a
simple discrete-time stochastic model has been presented. A part of our work is motivated by the model in [6]. However, our work differs from the work of [6] in the following respects:

- We have developed a simple stochastic model for (proposed) HnH system which combines two or more live streaming channels such that some of the peers are viewing the respective channel whereas the other peers are cooperating without watching the channel. In our work we have presented models for three different scenarios of HnH where two of them involves peers from viewing channel and cooperating channel.
- In the analysis of the model, our work is motivated by the model in [6] where interactions among peers only from the same viewing channel are considered. However, in our case we have our different views for the relative position of the playback pointers, ranges of overlapping pieces, etc.
- We have modeled single sub-stream buffering for the cooperating peers whereas the viewing peers play back all the pieces.
- In this model, each peer, instead of one, maintains two separate buffers in their memory: one for the viewing channel, one for the cooperated channel. Due to single sub-stream buffering, the cooperating peers maintain a different (preferably smaller) size of buffer for the cooperated channel.

III. DESCRIPTION OF THE StOCHASTIC MODEL

A. Problem Statement

In this work we develop a simple discrete-time stochastic model for Multi-Channel Live Streaming Systems, particularly, for the proposed HnH system with channels having small number of participants. This model investigates the fundamental characteristics, limitations and performance problems of P2P live streaming systems for such channels w.r.t. network parameters like number of neighbors, number of pieces present in the buffer etc. In order to solve the performance problems of such channels, we also propose a new way of forming overlay for live streaming channels having small number of participants.

B. Description of the System

In the (proposed) HnH system, a peer may have two types of neighbors: (i) those who are watching the same viewing channel and (ii) those who are cooperating peers of type(i) without viewing that channel. In this work, we refer to peers of type(i) as viewing peers and peers of type(ii) as cooperating peers. Figure 1 shows how the peers from two different types of channels communicate in the proposed HnH system. We identify three significant scenarios of interaction among the peers when we consider one channel (e.g., Ch-1):

- Viewing-to-Viewing(V2V)
  A peer from the viewing channel requests and downloads streaming contents from another peer which is also watching the same channel. In Figure 1, the solid directed line from peer A to peer B and from peer B to peer A indicate V2V interactions.
- Viewing-to-Cooperating(V2C)
  In this case, a peer from the viewing channel (e.g., peer A) requests and downloads streaming contents from a peer of a cooperating channel (e.g., peer D). In Figure 1, the solid line from peer A to peer D represents a V2C scenario.
- Cooperating-to-Viewing(C2V)
  In a C2V case, a peer from the cooperating channel (e.g., peer D) requests and downloads streaming contents from a peer from the viewing channel (e.g., peer D). It is to be noted that peer D does not watch the channel but buffers the streaming content in order to cooperate peers of ch-1. In Figure 1, the dotted line from peer D to peer A represents a C2V interaction.

In the present work, we assume that a cooperating peer from ch-2 does not buffer all the pieces of the stream rather it buffers only a sub-stream of ch-1. That is why it is reasonable to consider that a cooperating peer from ch-2 contacts and gets streaming contents for ch-1 only from the peers who are watching ch-1.

C. Assumptions

We made the following assumptions:

- In this stochastic model, we consider a multi-channel live streaming system. In such a system, there may be several channels which have a large number of participants whereas some of them might have a small number of participants. In this work, we only consider those channels having small number of participants.
In this work, it is assumed that the overlay for a channel is given and it follows the HnH approach for combining two or more channels. The details of the HnH approach is not covered here but it is assumed that, in a HnH system, a viewing peer of a DCSV channel has a sufficient number of neighbors.

We assume that each peer maintains a fixed number of neighbors, \( H \).

It is assumed that the download capacity of any peer is unlimited but the upload capacity of it is limited. We assume the upload capacity same for all the peers and it is further assumed as 1 piece/time slot.

In order to fetch the streaming contents before playback, each peer maintains a playback buffer in its memory. This buffer is assumed to be limited in size and each viewing peer maintains a buffer, \( L_s \), of same length.

Each cooperating peer maintains a buffer, \( L_c \) which keeps tracks of the same range of pieces as a \( L_s \) buffer does but contains only a sub-stream of that channel with the range of \( L_s \) sequences. Physically, it stores \( L_c/m \) (where \( m \) is the number of sub-streams at the source) pieces while it keeps track of \( L_c \) pieces, virtually.

In order to keep this simple, we assume \( L_c = L_s \).

For the playback, we assume that, playback rate is 1 piece/time slot. Thus, a playback pointer of a peer denotes the sequence number the piece that the peer is playing at that time slot. Whatever pieces are already played back are old pieces and whatever pieces are yet to be played are new pieces (also referred to as useful pieces if the sequences can be placed in the buffer in the current time slot).

If a viewing peer has \( j \) useful pieces in its buffer \( L_s \) then all the old pieces (i.e., \( L_s - j \)) are available in its buffer before they get replaced by newer pieces.

Similarly, if a cooperating peer has \( j \) useful pieces in its buffer \( L_c \) then all the \( L_c/m - j \) old pieces are available in its buffer before they are replaced by newer pieces; where \( m \) denotes the number of sub-streams at the source.

If a viewing peer has \( i \) useful pieces in its buffer, \( L_s \), and looking for \( L_s - i \) pieces in the current time slot, we assume all the \( L_s - i \) pieces have equal priority to be fetched from its neighbors.

The present work is done assuming the overlay to be in its steady state condition. It is very difficult for the peers to acquire streaming pieces at the very beginning as well as at the very end of the streaming session. Most of the uploading and downloading take place in the steady state (i.e., in the middle of the session) which can be defined as follows:

\[
N - L_s \geq t_s \geq L_s
\]

where, \( N \) denotes the total number of pieces of the streaming content, \( t_s \) denotes the playback pointer at source and \( L_s \) the length of buffer of the viewing peer and \( N >> L_s \).

### D. Inputs and Outputs

In a live streaming system, the media content is divided into smaller pieces/chunks for distribution. However, the number of total pieces is not known in advance (as opposed to the case in a file sharing system).

We introduce the following notations for input parameters:

**Parameters:**

- \( L_s \) → denotes the length of the buffer that a peer keeps for the channel it is watching currently
- \( L_c \) → denotes the length of the buffer that a peer keeps for the channel it is cooperating currently
- \( m_i \) → denotes the number of sub-streams at the source in Channel \( i \)
- \( t_s \) → denotes the playing time at the source server
- \( (t_s - 1) \) → denotes the earliest playback time at any peer
- \( (t_s - T) \) → denotes the latest playback time at any peer
- \( T \) → denotes the maximum delay for a peer to play a piece after the source has already played it.
- \( H \) → denotes the number of neighbors (it is assumed a fixed number of neighbors for all types of peers)

The proposed stochastic model will provide the probability of continuity which can be defined as, probability that a randomly picked peer in the overlay network that is watching or listening to a certain live streaming content in a specific time slot, would be able to play its desired piece at the same time slot. From there we will be able to gain various insights (e.g., effect of number of neighbors, effect of buffer length, on the probability of continuity).

**E. Objective**

The objective in this discrete-time stochastic model for Multi-Channel Live Streaming Systems is to find the probability of continuity, such that a randomly picked peer in the overlay network having a small number of participants (i.e., watching a live streaming content) in a specific time slot, would be able to play its desired piece in that specific time slot.

**F. Constraints**

We have the following constraints considered in the proposed discrete-time stochastic model for a HnH live streaming system:

- When a piece of streaming content (e.g., \( \text{piece}_i \)) arrives to a peer (e.g., peer A) it will be buffered only if \( i \geq t_A \) where \( t_A \) is the playback pointer of that peer. Otherwise, \( \text{piece}_i \) will be discarded.
- If a viewing peer (e.g., peer A) with playback pointer \( t_A \) has \( i \) useful pieces in its buffer \( L_s \) then the maximum amount of new pieces it can download and place in its buffer is \( (L_s - i) \).
- If a cooperating peer (e.g., peer D) has \( j \) useful pieces in its buffer \( L_c \) and interested to download only a particular sub-stream of the streaming channel, then the maximum amount of new pieces it can download and place in its buffer is \( \frac{L_c}{m} - j \), where \( m \) is the number of sub-streams at the source.
In general, we find the following two situations for relative playback pointers and the boundary of their streaming buffers.

1. Peer A is playing ahead of peer B and A’s playback pointer is within the boundary of the buffer of peer B and there is certain overlapping of useful pieces between A and its neighbor B.
2. Peer B is playing ahead of peer A and B’s playback pointer is within the boundary of the buffer of peer A and there is certain overlapping of useful pieces between A and its neighbor B.

These five general categories are presented in Figure 2. In the subsequent sections, these categories will be explored in details for the specific cases. Now we develop discrete-time stochastic models for each of the scenarios mentioned in the Description of the System.

### A. Viewing-to-Viewing (V2V) Peers’ Interactions

In this scenario, both the randomly picked peer (e.g., peer A) and its neighbor (e.g., peer B) are viewing peers. We assume, peer A has $i$ useful pieces in its buffer of size $L_s$ and peer B has $j$ useful pieces in its buffer of same size. At the beginning, we are interested to know the probability that A will be interested about the pieces present in the buffer of its randomly picked neighbor. Hence, we proceed as follows: We define $F^{V2V}(i,j)$ as the probability that peer A with $i$ useful pieces in its buffer, $L_s$, in the current time slot will not be interested in peer B, which has $j$ useful pieces in its buffer in the same time slot.

It is obvious that peer A would not be interested in the pieces...
of peer B if the buffer of peer A already contains or played back all the pieces that peer B has in its buffer at that time slot.

We calculate $F^{v2v}(i,j)$ for the following mutually five exclusive cases which are based on the categories presented in Figure 2:

$Case_1^{v2v}$: $t_s - T \leq t_B \leq t_A - L_s$
$Case_2^{v2v}$: $t_A + 2L_s - j \leq t_B \leq t_s - 1$
$Case_3^{v2v}$: $t_A - L_s + 1 \leq t_B \leq t_A$
$Case_4^{v2v}$: $t_A + 1 \leq t_B \leq t_A + L_s - 1$
$Case_5^{v2v}$: $t_A + L_s \leq t_B \leq t_A + 2L_s - j - 1$

At first, we explain the above mentioned cases with the help of diagrams. Then we calculate the probabilities for each of those cases, such that peer A will not be interested in the pieces of peer B.

$Case_1^{v2v}$: $t_s - T \leq t_B \leq t_A - L_s$:
Peer A has $i$ useful pieces in its buffer and peer B has $j$ useful pieces in its buffer and there is no overlapping of the boundaries of the buffers of A and B. As shown in Figure 3, it is clear that peer A has played back all the pieces those lie within the boundary of the buffer, $L_s$, of peer B and has no interest in those pieces of neighbor B.

We define probability $f_1^{v2v}(i,j)$ such that, in the current time slot, peer A is not interested in the pieces of peer B under the condition of $Case_1^{v2v}$.

$$f_1^{v2v}(i,j) = \sum_{t_A=t_s-T}^{t_s-1} \sum_{t_B=t_A-L_s}^{t_A-L_s} \frac{1}{T^2}$$  \hspace{1cm} (1)

$Case_2^{v2v}$: $t_A + 2L_s - j \leq t_B \leq t_s - 1$:
Peer A has $i$ useful pieces in its buffer and peer B has $j$ useful pieces and $(L_s - j)$ old pieces in its buffer and there is no overlapping of the boundaries of the buffers of A and B. As shown in Figure 4, the lowest numbered old piece in the buffer of peer B is greater than highest sequence number of piece that A can store in its buffer, $L_s$. Clearly, peer A and has no interest in those pieces of neighbor B.

We define probability $f_2^{v2v}(i,j)$ such that, in the current time slot, peer A is not interested in the pieces of peer B under the condition of $Case_2^{v2v}$.

$$f_2^{v2v}(i,j) = \sum_{t_A=t_s-T}^{t_s-1} \sum_{t_B=t_s-T}^{t_B=min\{t_s-1,t_A+2L_s-j\}} \frac{1}{T^2}$$ \hspace{1cm} (2)

$Case_3^{v2v}$: $t_A - L_s + 1 \leq t_B \leq t_A$:
Peer A has $i$ useful pieces in its buffer and peer B has $j$ useful pieces and among these $j$ pieces $x$ pieces are already present in the overlapping portion of the buffer of peer A. More specifically, whatever pieces peer B has in the overlapping portion of its buffer, peer A also has those pieces in its buffer. As shown in Figure 5, the remaining $(j - x)$ pieces are of no interest to A as those pieces are already played by peer A (and they are located in the region before the playback pointer of peer A).

We define probability $f_3^{v2v}(i,j)$ such that, in the current time slot, peer A is not interested in the pieces of peer B under the condition of $Case_3^{v2v}$.

$$f_3^{v2v}(i,j) = \sum_{t_A=t_s-T}^{t_s-1} \sum_{t_B=t_B=max\{t_s-1,t_A-L_s+1\}}^{t_B=max\{t_s-1,t_A-t_B\}} \sum_{x=max\{0,j-(t_A-t_B)\}}^{\min\{L_s-(t_A-t_B),i,j\}} \frac{L_s-(t_A-t_B)}{x-\frac{t_A-t_B}{j}} \times \frac{t_s-t_B}{t_B} \times \frac{t_s-x}{t_s-x} \times \frac{1}{T^2}$$ \hspace{1cm} (3)

$Case_4^{v2v}$: $t_A + 1 \leq t_B \leq t_A + L_s - 1$:
Peer A has $i$ useful pieces in its buffer and peer B has $j$ useful pieces and $L_s - j$ old pieces in its buffer. Among the $j$ pieces, B has $x$ pieces in the overlapping part of the useful pieces and peer A already have all those $x$ pieces in its buffer. Moreover, peer A has all the $L_s - j$ old pieces of peer B. As shown in Figure 6, the remaining $(j - x)$ pieces
Case where $y$ are of no interest to $A$ as those pieces are too new for peer $A$ to buffer.

We define probability $f^i_{42}(i, j)$ such that, in the current time slot, peer $A$ is not interested in the pieces of peer $B$ under the condition of Case $4^2$.

$$f^i_{42}(i, j) = \frac{1}{T^2} \sum_{t_A = t_A - T}^{t_A - 1} \sum_{t_B = \min\{t_B, t_A + L_s\}}^{t_B - 1} \text{min}\{t_A - [t_A + L_s - j - 1]\} \times \frac{1}{T^2} \sum_{x = \max\{0, t_B - t_A\}}^{t_B - t_A} \left(\frac{t_B - t_A}{j - x} \times \frac{t_B - t_A}{x} \times \frac{L_s - y}{i - y}\right) \times \frac{1}{T^2} \tag{4}$$

where $y = x + \min\{t_B - t_A, (L_s - j)\}$

Case $5^2$: $t_A + L_s \leq t_B \leq t_A + 2L_s - j - 1$:

Peer $A$ has $i$ useful pieces in its buffer and peer $B$ has $j$ useful pieces and $L_s - j$ old pieces in its buffer. There is no overlapping of useful pieces of their buffers. However, useful pieces of peer $A$ has overlapping with $y$ old pieces of peer $B$ and peer has all those $y$ pieces in its buffer. As shown in Figure 7, it is obvious that peer $A$ will not be interested about the $j$ useful pieces of peer $B$ as those pieces are too new for peer $A$ to buffer.

We define probability $f^i_{52}(i, j)$ such that, in the current time slot, peer $A$ is not interested in the pieces of peer $B$ under the condition of Case $5^2$.

$$f^i_{52}(i, j) = \frac{1}{T^2} \sum_{t_A = t_A - T}^{t_A - 1} \sum_{t_B = \min\{t_B, t_A + L_s\}}^{t_B - 1} \text{min}\{t_A - [t_A + L_s - j - 1]\} \times \frac{1}{T^2} \sum_{x = \max\{0, t_B - t_A\}}^{t_B - t_A} \left(\frac{t_B - t_A}{j - x} \times \frac{t_B - t_A}{x} \times \frac{L_s - y}{i - y}\right) \times \frac{1}{T^2} \tag{5}$$

where $y' = t_A + L_s - [t_B - (L_s - j)]$

Now, $F^v_{42}(i, j)$ can be expressed as the sum of the probabilities of the above five mutually exclusive cases.

$$F^v_{42}(i, j) = f^i_{42}(i, j) + f^i_{22}(i, j) + f^i_{32}(i, j) + f^i_{42}(i, j) + f^i_{52}(i, j) \tag{6}$$

At any given time slot, if peer $B$ has a piece or some pieces that peer $A$ is interested in, peer $A$ will send a request to peer $B$. We recall that, in this work, we assume any random peer in a network has, $H$, fixed number of neighbors and it possesses a buffer, $L_s$, of fixed length. We also assume that at any time slot if any randomly picked peer (e.g. peer $B$) receives more than one request, only one of them will be fulfilled randomly by peer $B$.

When a randomly picked peer in the network has $i$ pieces (i.e., peer $A$), then we define the maximum number of requests which can be sent by that peer (i.e., $A$) to its neighbors, $s_{max}$ as:

$$s_{max} = \min\{H, (L_s - i)\};$$

where $(L_s - i)$ is number of pieces that peer $A$ wants to download at that specific time slot.

Now, we define the probability, $U^v_{42}$, that a randomly selected peer (e.g., peer $A$) which has $i$ useful pieces in its buffer in a specific time slot, will be interested to get a piece from one of its randomly selected neighbors (e.g., peer $B$) as follows:

$$U^v_{42}(i, j) = \sum_{j = 0}^{L_s} \left(1 - F^v_{42}(i, j) \times P^v_{42}(j) \right) \tag{7}$$

where $P^v_{42}$ is the probability that a random peer in the network has $j$ useful pieces.

In the next step, we find the probability that a randomly selected peer in the network (e.g. peer $A$) sends $k$ requests to its neighbors.

We define $F^v_{42}(H, i', k)$, as the probability that a randomly selected peer which has $i$ useful pieces in its buffer, in a specific time slot, and it is looking for $i'$, where $i' = L_s - i$, pieces in that time slot, sends $k$ requests to its neighbors. It is obvious that $0 \leq k \leq \min\{L_s - i, H\}$.

$F^v_{42}(H, i', k)$ is a recursive function which can be calculated as follows:

$$F^v_{42}(h, i', k) = U^v_{42}(i, k) \times F^v_{42}(h - 1, i' - 1, k - 1) + (1 - U^v_{42}(i, k)) \times F^v_{42}(h - 1, i', k) \tag{8}$$
where \( H \geq h \geq 0, i = 0, 1, ..., L_s \text{ and } i' = 0, ..., L_s - i \) and \( k = 0, ..., \min \{ h, i' \} \)

The first portion in equation (8) (i.e., \( U^v_{i'} \times F^v(0, i' - 1, k) \)) indicates that the peer is interested about one piece from one of the \( m \) neighbors with probability \( U^v_{i'} \). Hence, it is looking for one less piece (i.e., \( i' - 1 \)) from one less neighbors (i.e., \( h - 1 \)) and sending one less request (i.e., \( k - 1 \)).

The second portion of the equation (8) (i.e., \((1 - U^v_{i'}) \times F^v(1, i' - 1, k)\)) indicates that the peer is not interested about pieces from one of its \( m \) neighbors with probability \((1 - U^v_{i'})\). Hence, it is looking for all the pieces (i.e., \( i' \)) but from one less than total neighbors (i.e., \( h - 1 \)) while sending same number of requests (i.e., \( k \)).

We have the following two initial conditions for equation (8):

\[
F^v(h, i', 0) = 1; h = 0, k = 0, i = 0, 1, ..., L_s \\
F^v(h, 0, i') = 0; H \geq h \geq 0, i = 0, 1, ..., L_s \\
\quad ; i' = 0, ..., \min \{ h, i' \}; k \neq 0
\]

We can derive \( F(h, i', k) \) by solving the following \( H \) equations:

\[
F^v(h, 0, 0) = 0 \\
F^v(h, 1, 0) = U^v_{i'} \times F^v(0, i' - 1, k) \\
\quad + (1 - U^v_{i'}) \times (1 - F^v(0, i', k)) \\
F^v(h, 2, 0) = U^v_{i'} \times F^v(1, i' - 1, k) \\
\quad + (1 - U^v_{i'}) \times F^v(1, i', k) \\
F^v(h, 3, 0) = U^v_{i'} \times F^v(2, i' - 1, k) \\
\quad + (1 - U^v_{i'}) \times F^v(2, i', k) \\
\ldots \\
F^v(h, 0, i') = 0; H \geq h \geq 0, i = 0, 1, ..., L_s \\
\quad ; i' = 0, ..., \min \{ h, i' \}; k \neq 0
\]

Next, we define \( \overline{K}^v \), as the average number of requests that a randomly picked peer, which has \( i \) useful pieces and is looking for \( (L_s - i) \) pieces in a specific time slot, will send to its neighbors. It can be derived by calculating the expected value of \( F^v(h, i', k) \), as below:

\[
\overline{K}^v = \frac{\text{Expected value}\left[ F^v(h, i', k) \right]}{\min \{ i', H \}} = \sum_{k=0}^{\min \{ i', H \}} k \times \overline{F}^v(h, i', k) \tag{12}
\]

where \( i = 0, 1, ..., L_s \) and \( i' = 0, ..., L_s - i \) and \( k = 0, ..., \min \{ H, i' \} \)

At any given time slot, we define \( \overline{k}^v \), as the average number of requests that any randomly picked peer sends to its neighbors at any given time slot. \( \overline{k}^v \) can be expressed as:

\[
\overline{k}^v = \sum_{i'=0}^{L_s} (P^v_{i'} \times \overline{F}^v_{i'}) \tag{13}
\]

We also define \( \overline{X}^v \), as the average number of requests that a randomly selected peer (e.g., peer B) receives from its neighbors in addition to the request received from peer A, as below:

\[
\overline{X}^v = \frac{(H - 1) \times \overline{k}^v}{H} \tag{14}
\]

Now, we introduce \( Q^v \), as the probability that a randomly selected peer (e.g., peer B) fulfills a specific request (e.g., request from peer A) among all the requests that it received. \( Q^v \) can be expressed as follows:

\[
Q^v = \frac{1}{1 + \overline{X}^v} \tag{15}
\]

where \( i = 0, 1, ..., L_s \) and \( i' = 0, ..., (L_s - i) \)

Next, we define \( r^v_{i, n} \), as the probability that a peer, which has \( i \) useful pieces at a given time slot, downloads \( n \) pieces in the same time slot. We recall that, the maximum number of requests that can be sent by this peer will be:

\[
s^v_{\text{max}} = \min \{ L_s - i, H \}
\]

It is also obvious that the number of pieces which can be downloaded by this peer at any given time slot is a Binomial random variable with parameters \( k \) and \( Q^v \) expressed as follows:

\[
r^v_{i, n} = \sum_{k=n}^{\min \{ L_s - i, H \}} \left( \frac{F^v(H, L_s - i, k)}{n} \right) \times ((1 - Q^v)^{k-n}) \tag{16}
\]

where \( i = 0, 1, ..., L_s \) and \( i' = 0, ..., (L_s - i) \), \( s^v_{\text{max}} = \min \{ L_s - i, H \} \) and \( n = 0, 1, ..., s^v_{\text{max}} \)

In our analysis, we assume that in a specific time slot, in a randomly picked peer’s buffer in the network, desired pieces, which are scheduled to be downloaded in this time slot, gets downloaded and stored in the buffer by that peer is uniformly distributed.

We define \( M^v \) as the probability that a randomly picked peer in the network with the buffer length of size \( L_s \) and \( i \) useful pieces, which are uniformly distributed, in its buffer in a specific time slot, can play a piece in this time slot, would be expressed as follows:

\[
M^v = \frac{i}{L_s} \tag{17}
\]

where \( i = 0, 1, ..., L_s \)

We assumed that the system is in the steady state, hence, the peer distribution does not change with time.

In Figure 8 if we consider any particular state (e.g., \( i \)th state) and all the incoming and outgoing transitions to and from
state \( i \), we find the following relationship is true as shown in equation 18:

\[
(M_{i+1}^{v2v} \times P_i^{v2v}) + \left( \sum_{k=0}^{\min(H_i)} (r_{i-k,k}^{v2v} \times P_i^{v2v}) \right) - \left( \sum_{k=1}^{L_s} (r_{i,k}^{v2v} \times P_i^{v2v}) \right) - (M_i^{v2v} \times P_i^{v2v}) = 0 \tag{18}
\]

here \( i = 0, 1, \ldots, L_s \) and \( \sum_{i=0}^{L_s} (P_i^{v2v}) = 1 \)

(It is difficult to find a closed form of solution for equation 18 in order to get peer distribution \( \{P_i^{v2v}\} \). However, it is possible to solve equation (18) numerically)

Then, we define \( P_{cont}^{v2v} \), as the probability that a randomly picked peer in the network that is watching or listening to a certain live streaming content in a specific time slot, would be able to play its desired piece at this time slot. It can be expressed as follows:

\[
P_{cont}^{v2v} = \sum_{i=0}^{L_s} P_i^{v2v} \times M_i^{v2v} \tag{19}
\]

We define \( d_i^{v2v} \), as the average download rate of a peer that has \( i \) useful pieces in its buffer of size \( L_s \) in a specific time slot as follows:

\[
d_i^{v2v} = \min\{H, L_s-i\} \sum_{k=0}^{\min\{H, L_s-i\}} (k \times r_{i,k}^{v2v}) \tag{20}
\]

where \( i = 0, 1, \ldots, L_s \)

B. Viewing-to-Cooperating (V2C) Peers’ Interactions

In this scenario, the randomly picked peer (e.g., peer A) is a viewing peer and its random neighbor (e.g., peer D) is a cooperating peer. We assume, peer A has \( i \) useful pieces in its buffer of size \( L_s \) and peer D has \( j \) useful pieces in its buffer \( L_c \). As in the previous scenarios, we are interested to know the probability that A will be interested about the pieces present in the buffer of its randomly picked neighbor D. Hence, we proceed as follows:

We define \( F_{1}^{v2c}(i, j) \) as the probability that peer A with \( i \) useful pieces in its buffer in the current time slot will not be interested in peer D, which has \( j \) useful pieces in its buffer in the same time slot. It is obvious that peer A would not be interested in peer D’s pieces if peer A contains all the pieces that peer D has in its buffer at that time. It has been assumed that as a cooperating neighbor, peer D does not buffer all the pieces, rather it takes care of a specific sub-stream. In order to derive \( F_{1}^{v2c}(i, j) \), we consider five mutually exclusive cases depending on the situation of the playtime, \( t_D \), of neighbor D and the playtime, \( t_A \), of peer A. We have the following 5 mutually exclusive cases:

\[
Case_1^{v2c} \colon t_s - T \leq t_D \leq t_A - L_c
\]

\[
Case_2^{v2c} \colon t_A + L_s + m \times \left( \frac{L_c}{m} - j \right) \leq t_D \leq t_s - 1
\]

\[
Case_3^{v2c} \colon t_A - L_c + 1 \leq t_D \leq t_A
\]

\[
Case_4^{v2c} \colon t_A + 1 \leq t_D \leq t_A + L_s - 1
\]

\[
Case_5^{v2c} \colon t_A + L_s \leq t_D \leq t_A + L_s + m \times \left( \frac{L_c}{m} - j \right) - 1
\]

We calculate the probabilities such that peer A will not be interested in the pieces in the buffer of peer D.

\[
Case_3^{v2c} \colon t_s - T \leq t_D \leq t_A - L_c
\]

Peer A has \( i \) useful pieces in its buffer and peer B has \( j \) useful pieces in its buffer, the playback pointer of peer A is already ahead of the playback pointer of its neighbor and there is no overlapping of the boundaries of the buffers of A and B. As shown in Figure 9, it is clear that peer A has played back all the pieces those lie within the boundary of the buffer, \( L_s \), of peer D and has no interest in those pieces of neighbor D.

We define probability \( f_{1}^{v2c}(i, j) \) such that, in the current time slot, peer A is not interested in the pieces of peer D under the condition of \( Case_1^{v2c} \).

\[
f_{1}^{v2c}(i, j) = \sum_{t_s-t_D=t_A-L_s}^{t_s-t_D=t_A-L_c} \frac{1}{T^2} \tag{21}
\]

\[
Case_2^{v2c} \colon t_A + L_s + m \times \left( \frac{L_c}{m} - j \right) \leq t_D \leq t_s - 1
\]

\[
Case_2^{v2c} \text{ has been depicted in Figure: 10. We find that the playback pointer of the cooperating neighbor too far from peer A. However, peer A has no interest in the old pieces of peer D as those pieces are too new for peer A. We define probability } f_{2}^{v2c}(i, j) \text{ such that, in the current time slot, peer A is not interested in the pieces of peer D under the condition}
\]
of Case$_2^{v2c}$:

$$f_2^{v2c}(i, j) = \frac{\sum_{t_A=t_s-T}^{t_s-1} \sum_{t_D=t_A+L_s+(1/m)(L_c/m-j)}^{t_s-1} 1}{T^2}$$

Case$_3^{v2c}[t_A - L_c + 1 \leq t_D \leq t_A]$:
Viewing peer A has $i$ useful pieces in its buffer and cooperating peer D has $j$ useful pieces and among these $j$ pieces $x$ pieces are already present in the overlapping portion of the buffer of peer A. More specifically, whatever pieces peer D has (corresponding to its sub-stream) in the overlapping portion of its buffer, peer A also has those pieces in its buffer. As shown in Figure 11, the remaining $(j-x)$ pieces are of no interest to A as those pieces are already played by peer A (and they are located in the region before the playback pointer of peer A).

We define probability $f_3^{v2c}(i, j)$ such that, in the current time slot, peer A is not interested in the pieces of peer D under the condition of Case$_3^{v2c}$.

$$f_3^{v2c}(i, j) = \frac{\sum_{t_A=t_s-T}^{t_s-1} \sum_{t_D=t_A+L_s+(1/m)(L_c/m-j)}^{t_s-1} \min[L_c - (t_A-t_D), i]}{T^2}$$

$$\times \left( \frac{L_c}{m} \right) \times \left( \frac{L_c - i}{m} \right) \times \frac{1}{T^2}$$

where $y = x + \min \left\{ \frac{(t_D-t_A)}{m}, \frac{L_c}{m} \right\}$.

Case$_4^{v2c}[t_A + 1 \leq t_D \leq t_A + L_s - 1]$:
Viewing peer A has $i$ useful pieces in its buffer and cooperating peer D has $j$ useful pieces and $L_c/m - j$ old pieces in its buffer. Among the $j$ pieces, D has $x$ pieces in the overlapping part of the useful pieces and peer A already have all those $x$ pieces in its buffer. Moreover, peer A has all the $L_c/m - j$ old pieces of D. As shown in Figure 12, the remaining $(j-x)$ pieces are of no interest to A as those pieces are too new for peer A to buffer.

We define probability $f_4^{v2c}(i, j)$ such that, in the current time slot, peer A is not interested in the pieces of peer D under the condition of Case$_4^{v2c}$.

$$f_4^{v2c}(i, j) = \frac{\sum_{t_A=t_s-T}^{t_s-1} \sum_{t_D=t_A+L_s+(1/m)(L_c/m-j)}^{t_s-1} \min \left\{ t_s-1, \frac{(t_A+L_s)}{m} \right\} - \left( \frac{L_c}{m} - j \right) - 1}{T^2}$$

$$\times \left( \frac{L_c}{m} \right) \times \left( \frac{L_c - i}{m} \right) \times \frac{1}{T^2}$$

$$\times \left( \frac{L_c - (t_D-t_A)}{m} \right) \times \left( \frac{L_c - (t_D-t_A)}{m} \right) \times \frac{1}{T^2}$$

where $y = x + \min \left\{ \frac{(t_D-t_A)}{m}, \frac{L_c}{m} \right\}$.
can be sent by that peer (i.e., A) to its neighbors, peer D. We also assume that at any time slot if any randomly that peer A is interested in, peer A will send a request to At any given time slot, if peer D has a piece (or some pieces) buffer (e.g. peer A), will be interested to get a piece from one peer which has i useful pieces in a specific time slot in its

Fig. 13. Case $5^{2c}$: $t_A + L_s \leq t_D \leq t_A + L_s + m \times (\frac{L_c}{m} - j) - 1$

( peer D is playing much ahead of peer A such that only old pieces of D (i.e., $L_s \leq m - j$) pieces have overlapping with the buffer of peer A)

peer D as those pieces are too new for peer A to buffer.

We define probability $f_5^{2c}(i, j)$ such that, in the current time slot, peer A is not interested in the pieces of peer D under the condition of Case $5^{2c}$.

$$f_5^{2c}(i, j) = \sum_{t_A = t_s - T}^{t_s - 1} \sum_{t_D = \min\{t_s - 1, t_A + L_s\}} \frac{(L_s - y')}{L_s} \times \frac{1}{T^2}$$

where $y' = t_A + L_s - [t_D - m \times (\frac{L_c}{m} - j)]$

Now, $F^{2c}(i, j)$ can be expressed as the sum of the probabilities of the above mentioned mutually exclusive cases.

$$F^{2c}(i, j) = f_1^{2c}(i, j) + f_2^{2c}(i, j) + f_3^{2c}(i, j) + f_4^{2c}(i, j) + f_5^{2c}(i, j)$$

At any given time slot, if peer D has a piece (or some pieces) that peer A is interested in, peer A will send a request to peer D. We also assume that at any time slot if any randomly picked peer (e.g. peer D) receives more than one request, only one of them will be fulfilled randomly by peer D. When a randomly picked peer in the network has i pieces (i.e., peer A), then we define the maximum number of requests which can be sent by that peer (i.e., A) to its neighbors, $s_{\text{max}}^{2c}$ as:

$$s_{\text{max}}^{2c} = \min\{H, (L_s - i)\}$$

where $(L_s - i)$ is number of pieces that peer A wants to download at that specific time slot.

We also need to find, the probability that a randomly selected peer which has i useful pieces in a specific time slot in its buffer (e.g. peer A), will be interested to get a piece from one of its randomly selected neighbors. We define this probability as follows:

$$U_{i}^{2c} = \sum_{j=0}^{L_c/m} (1 - F_{j}^{2c}(i, j)) \times P_{j}^{2c}$$

where $P_{j}^{2c}$ is the probability that a random peer in the network has j useful pieces.

In the next step, we need to find the probability that a randomly selected peer in the network (e.g. peer A) sends k requests to its neighbors. We define $F_{i}^{2c}(H, i', k)$, as the probability that a randomly selected peer which has i useful pieces, in a specific time slot, in its buffer and it is looking for $i'$, where $i' = L_s - i$, pieces in that time slot, sends k requests to its neighbors. It is obvious that $0 \leq k \leq \min\{L_s - i, H\}$, where $F_{i}^{2c}(H, i', k)$ is a recursive function which can be calculated as follows:

$$F_{i}^{2c}(H, i', k) = U_{i}^{2c} \times F_{i}^{2c}(H - 1, i' - 1, k - 1) + (1 - U_{i}^{2c}) \times F_{i}^{2c}(H - 1, i', k)$$

(28)

where $H \geq h \geq 0, i = 0, 1, ..., L_s$ and $i' = 0, ..., L_s - i$ and $k = 0, ..., \min\{h, i'\}$

We have the following initial conditions:

$$F_{i}^{2c}(h, i', k) = 1; h = 0, k = 0, i = 0, 1, ..., L_s, \quad i' = 0, ..., \min\{h, i'\}$$

(29)

$$F_{i}^{2c}(h, i', k) = 0; H \geq h \geq 0, i = 0, 1, ..., L_s, \quad i' = 0, ..., \min\{h, i'\}; k \neq 0$$

(30)

It can be seen that we can derive $F_{i}^{2c}(h, i', k)$ by solving the following $H$ equations, as they are shown below:

$$F_{i}^{2c}(0, i', k) = 0$$

$$F_{i}^{2c}(1, i', k) = U_{i}^{2c} \times F_{i}^{2c}(0, i' - 1, k - 1) + (1 - U_{i}^{2c}) \times F_{i}^{2c}(0, i', k)$$

$$F_{i}^{2c}(2, i', k) = U_{i}^{2c} \times F_{i}^{2c}(1, i' - 1, k - 1) + (1 - U_{i}^{2c}) \times F_{i}^{2c}(1, i', k)$$

$$F_{i}^{2c}(3, i', k) = U_{i}^{2c} \times F_{i}^{2c}(2, i' - 1, k - 1) + (1 - U_{i}^{2c}) \times F_{i}^{2c}(2, i', k)$$

$$\ldots$$

$$F_{i}^{2c}(H - 1, i', k) = U_{i}^{2c} \times F_{i}^{2c}(H - 2, i' - 1, k - 1) + (1 - U_{i}^{2c}) \times F_{i}^{2c}(H - 2, i', k)$$

$$F_{i}^{2c}(H, i', k) = U_{i}^{2c} \times F_{i}^{2c}(H - 1, i' - 1, k - 1) + (1 - U_{i}^{2c}) \times F_{i}^{2c}(H - 1, i', k)$$

(31)

Next, we define $\overline{K}_{i'v}^{2c}$, as the average number of requests that a randomly picked peer, which has i useful pieces and is looking for $(L_s - i)$ pieces in a specific time slot, will send to its neighbors. This can be derived by calculating the expected value of $F_{i}^{2c}(h, i', k)$, as below:

$$\overline{K}_{i'v}^{2c} = \sum_{k=0}^{\min\{h, i'\}} k \times F_{i}^{2c}(h, i', k)$$

(32)
where $i = 0, 1, ..., L_s$ and $i' = 0, ..., L_s - i$ and $k = 0, ..., \min \{H, i'\}$.

At any given time slot, we define $K^{v2c}$, as the average number of requests that any randomly picked peer sends to its neighbors at any given time slot. $K^{v2c}$ can be expressed as:

$$K^{v2c} = \sum_{i'=0}^{L} (P_{i'} \times K_i') \tag{33}$$

We also define $\bar{X}^{v2c}$, as the average number of requests that a randomly selected peer (e.g., peer D) receives from its neighbors in addition to the request received from peer A, as below:

$$\bar{X}^{v2c} = \frac{(H-1) \times K^{v2c}}{H} \tag{34}$$

Now, we introduce $Q^{v2c}$, as the probability that a randomly selected peer (e.g., peer D) fulfills a specific request (e.g., request from peer A) among all the requests that it received. $Q^{v2c}$ can be expressed as follows:

$$Q^{v2c} = \frac{1}{1 + \bar{X}^{v2c}} \tag{35}$$

where $i = 0, 1, ..., L_s$ and $i' = 0, ... (L_s - i)$

Next, we define $r_{i,n}^{v2c}$, as the probability that a peer, which has $i$ useful pieces at a given time slot, downloads $n$ pieces in the same time slot. We recall that, the maximum number of requests that can be sent by this peer will be:

$s^{v2c}_{\text{max}} = \min \{L_s - i, H\}$.

It is also obvious that the number of pieces which can be downloaded by this peer at any given time slot is a Binomial random variable with parameters $k$ and $Q^{v2c}$ expressed as follows:

$$r_{i,n}^{v2c} = \sum_{k=n}^{\min \{L_s - i, H\}} \left( F^{v2c}(H, L_s - i, k) \binom{k}{n} \right) \times \left( (Q^{v2c})^n \times (1 - Q^{v2c})^{k-n} \right) \tag{36}$$

where $i = 0, 1, ..., L_s$ and $i' = 0, ... (L_s - i)$, $s^{v2c}_{\text{max}} = \min \{L_s - i, H\}$ and $n = 0, 1, ..., s^{v2c}_{\text{max}}$

In our analysis, we assume that in a specific time slot, in a randomly picked peer’s buffer in the network, desired pieces, which are scheduled to be downloaded in this time slot, gets downloaded and stored in the buffer by that peer is uniformly distributed.

We define $M^{v2c}$ as the probability that a randomly picked peer in the network with the buffer length of size $L_s$ and $i$ useful pieces, which are uniformly distributed, in its buffer in a specific time slot, can play a piece in this time slot, would be expressed as follows:

$$M^{v2c} = \frac{i}{L_s} \tag{37}$$

where $i = 0, 1, ..., L_s$.

We assumed that the system is in the steady state, hence, the peer distribution does not change with time. The following is true in this situation:

In Figure 14 if we consider any particular state (e.g., $i$th state) and all the incoming and outgoing transitions to and from state $i$, we find the following relationship is true as shown in equation 38:

$$\left( M^{v2c}_{i+1} \times P^{v2c}_{i+1} \right) + \sum_{k=0}^{\min \{H, i\}} \left( r_{i-k,k}^{v2c} \times P^{v2c}_{i-k} \right) - \left( \sum_{k=1}^{L_s-i} \left( r_{i,k}^{v2c} \times P^{v2c}_{i-k} \right) \right) = 0 \tag{38}$$

where $i = 0, 1, ..., L_s$ and $\sum_{i=0}^{L_s} (P^{v2c}_{i}) = 1$

(It is difficult to find a closed form of solution but it can be solved numerically)

Then, we define $P^{v2c}_{\text{cont}}$, as the probability that a randomly picked peer in the network that is watching or listening to a certain live streaming content in a specific time slot, would be able to play its desired piece at this time slot. It can be expressed as follows:

$$P^{v2c}_{\text{cont}} = \sum_{i=0}^{L_s} P^{v2c}_{i} \times M^{v2c}_{i} \tag{39}$$

We define $d^{v2c}_{i}$, as the average download rate of a peer that has $i$ useful pieces in its buffer of size $L_s$ in a specific time slot as follows:

$$d^{v2c}_{i} = \sum_{k=0}^{\min \{H, L_s-i\}} \left( k \times r_{i,k}^{v2c} \right) \tag{40}$$

where $i = 0, 1, ..., L_s$

C. Cooperating-to-Viewing (C2V) Peers’ Interactions

In this scenario, the randomly picked peer (e.g., peer D) is a cooperating peer and its random neighbor (e.g., peer A)
is a viewing peer. We assume, peer D has \(i\) useful pieces in its buffer of size \(L_c\) and peer A has \(j\) useful pieces in its buffer \(L_s\). As in the previous scenarios, we are interested to know the probability that D will be interested about the pieces present in the buffer of its randomly picked neighbor A. Hence, we proceed as follows:

We define \(F_{c2v}(i,j)\) as the probability that peer D with \(i\) useful pieces in its buffer in the current time slot will not be interested in peer A, which has \(j\) useful pieces in its buffer in the same time slot. It is obvious that peer D would not be interested in peer A’s pieces if peer D contains all the pieces that peer A has in its buffer at that time. It has been assumed that peer D does not look for all the pieces, rather it takes care that peer A has in its buffer at that time. It is obvious that peer D would not be interested in peer A, which has useful pieces in its buffer in the current time slot will not be interested in piece A. Hence, we proceed as follows:

We define \(F_{c2v}(i,j)\) as the probability that peer D with \(i\) useful pieces in its buffer in the current time slot will not be interested in peer A, which has \(j\) useful pieces in its buffer in the same time slot. It is obvious that peer D would not be interested in peer A’s pieces if peer D contains all the pieces that peer A has in its buffer at that time. It has been assumed that peer D does not look for all the pieces, rather it takes care of a specific sub-stream only. In order to derive \(F_{c2v}(i,j)\), we consider five mutually exclusive cases depending on the situation of the play time, \(t_A\), of neighbor A and the playtime, \(t_D\), of peer D. As compared to the previous two scenarios, the situation is different here as the peer D is looking for only pieces corresponding to a sub-stream from its neighbor A.

\[
\begin{align*}
\text{Case}_1^{c2v}: & \quad t_s - T \leq t_A \leq t_D - L_s \\
\text{Case}_2^{c2v}: & \quad t_D + L_c + L_s - j \leq t_A \leq t_s - 1 \\
\text{Case}_3^{c2v}: & \quad t_D - L_c - 1 \leq t_A \leq t_D \\
\text{Case}_4^{c2v}: & \quad t_D - 1 \leq t_A \leq t_D + L_c - 1 \\
\text{Case}_5^{c2v}: & \quad t_D + L_c \leq t_A \leq t_D + L_c + L_s - j - 1 \\
\end{align*}
\]

We calculate the probabilities for the above five cases such that peer D will not be interested in the pieces in the buffer of peer A.

\[
\text{Case}_1^{c2v}[t_s - T \leq t_A \leq t_D - L_s]: \\
\text{Cooperating peer D has } i \text{ useful pieces in its buffer and peer A has } j \text{ useful pieces in its buffer and there is no overlapping of the boundaries of the buffers of A and D. As shown in Figure 15, we find that the playback pointer of peer D is already ahead of the playback pointer of its neighbor A. However, peer D has no interest in the (old or new) pieces of peer A as those pieces are too new for peer D. We define probability } f_2^{c2v}(i,j) \text{ such that, in the current time slot, peer D is not interested in the pieces of peer A under the condition of } \text{Case}_2^{c2v}. \\
\]

\[
f_2^{c2v}(i,j) = \sum_{t_D = t_s - T}^{t_s - 1} \sum_{t_A = t_s - T}^{t_s - 1} \frac{1}{T^2}
\]

(42)

\[
\text{Case}_2^{c2v}[t_D - L_s + 1 \leq t_A \leq t_D]: \\
\text{Peer D has } i \text{ useful pieces in its buffer and peer A has } j \text{ useful pieces and among these } j \text{ pieces } x \text{ (common) pieces are already present in the overlapping portion of the buffer of peer A. Peer D has all those pieces } x \text{ in its buffer. As shown in Figure 17, the remaining } (j - x) \text{ pieces are of no interest to D as those pieces are already played by peer D (and they are located in the region before the playback pointer of peer D).} \\
\]

We define probability \(f_3^{c2v}(i,j)\) such that, in the current time slot, peer D is not interested in the pieces of peer A under the condition of \(\text{Case}_3^{c2v}\).
**Fig. 18.** *Case* $c_{12}^v$: $t_D \leq t_A \leq t_D + L_c - 1$ (Peer A is playing before peer D and they have some overlapping pieces (i.e., $x$ pieces))

**Fig. 19.** *Case* $c_{13}^v$: $t_D + L_c \leq t_A \leq t_D + L_c + L_s - j - 1$ (Peer A is playing much ahead of peer D such that only old pieces of A (i.e., $(L_s-j)$ pieces) have overlapping with the buffer of peer D)

\[
\begin{align*}
\text{Case}_{12}^v & \quad [t_D + 1 \leq t_A \leq t_D + L_c - 1]: \\
\text{Cooperating peer D has } i \text{ useful pieces in its buffer and viewing peer A has } j \text{ useful pieces and } L_s - j \text{ old pieces in its buffer. Among these } j \text{ pieces, A has } x \text{ pieces in the overlapping part of the useful pieces and peer D already have all those } x \text{ pieces in its buffer. Moreover, peer D has all the } L_s/m - j \text{ old pieces (i.e., only corresponding to the sub-stream) of A. As shown in Figure 18, the remaining (} j - x \text{) pieces are of no interest to D as those pieces are too new for peer D to buffer.}
\end{align*}
\]

*Case* $c_{12}^v$ has been depicted in Figure: 18.

\[
\begin{align*}
\text{Case}_{13}^v & \quad [t_D + L_c \leq t_A \leq t_D + L_c + L_s - j - 1]: \\
\text{Cooperating peer D has } i \text{ useful pieces in its buffer and the viewing peer A has } j \text{ useful pieces and } L_s - j \text{ old pieces in its buffer. There is no overlapping of useful pieces of their buffers. However, useful pieces of peer D has overlapping with } y \text{ old pieces of peer A and peer D has all those } y/m \text{ pieces in its buffer. As shown in Figure 13, it is obvious that peer D will not be interested about the } j \text{ useful pieces of peer A as those pieces are too new for peer D to buffer.}
\end{align*}
\]

We define probability $f_{13}^v(i,j)$ such that, in the current time slot, peer D is not interested in the pieces of peer A under the condition of *Case* $c_{13}^v$. 

\[
\begin{align*}
\text{where } y = x + (1/m) \times \min \{((t_A - t_D), (L_s - j))\}
\end{align*}
\]
\[ f_{2v}^i(i, j) = \sum_{t_d = t_A}^{t_A - T} \min \{ t_s - 1, (t_d + L_c) - L_s + 1 \} \sum_{t_s = t_d}^{t_s - T} \left( \frac{L_c}{m} \right) \left( \frac{y'}{m} - \frac{y}{m} \right) \times \frac{1}{T^2} \]

(45)

where \( y' = t_d + L_c - [t_A - (L_s - j)] \).

Now, \( F^{c2v}(i, j) \) can be expressed as the sum of the probabilities of the five mutually exclusive cases mentioned in \( Case^{c2v}_1 \) through \( Case^{c2v}_5 \).

\[ F^{c2v}(i, j) = f_1^{c2v}(i, j) + f_2^{c2v}(i, j) + f_3^{c2v}(i, j) \]
\[ + f_4^{c2v}(i, j) + f_5^{c2v}(i, j) \]

(46)

At any given time slot, if peer A has a piece (or some pieces corresponding to the sub-stream of peer D) that peer D is interested in, peer D will send a request to peer A. We recall that, in this work, we assume any random cooperating peer in this network also has \( H \) fixed number of neighbors it possesses a buffer \( L \) of fixed length. We also assume that at any time slot if any randomly picked peer (e.g., peer A) receives more than one request, only one of them will be fulfilled randomly by peer A. When a randomly picked peer in the network has \( i \) pieces (i.e., peer D), then we define the maximum number of requests which can be sent by that peer (i.e., D) to its neighbors, \( s^{c2v}_{max} \) as:

\[ s^{c2v}_{max} = \min \{ H, \frac{L_c}{m} - i \} \] where \( \frac{L_c}{m} - i \) is number of pieces that peer D wants to download at that specific time slot.

We also need to find the probability that a randomly selected peer which has \( i \) useful pieces in a specific time slot in its buffer (e.g., peer D), will be interested to get a piece from one of its randomly selected neighbors. We define this probability as follows:

\[ U^{c2v}_i = \sum_{j=0}^{L_c/m} (1 - F^{c2v}(i, j)) \times P^{c2v}_j \]

(47)

where \( P^{c2v}_j \) is the probability that a random peer in the network has \( j \) useful pieces.

In the next step, we need to find the probability that a randomly selected peer in the network (e.g., peer D) sends \( k \) requests to its neighbors. We define \( F^{c2v}(H, i', k) \), as the probability that a randomly selected peer which has \( i \) useful pieces, in a specific time slot, in its buffer and it is looking for \( i' \), where \( i' = \frac{L_c}{m} - i \), pieces in that time slot, sends \( k \) requests to its neighbors. It is obvious that \( 0 \leq k \leq \min \{ \frac{L_c}{m} - i, H \} \).

\[ F^{c2v}(H, i', k) \] is a recursive function which can be calculated as follows:

\[ F^{c2v}(H, i', k) = U^{c2v}_i \times F^{c2v}(H - 1, i' - 1, k - 1) \]
\[ + (1 - U^{c2v}_i) \times F^{c2v}(H - 1, i', k) \]

(48)

where \( H \geq h \geq 0, i = 0, 1, ..., \frac{L_c}{m} \) and \( i' = 0, ..., \frac{L_c}{m} - i \) and \( k = 0, ..., \min \{ h, i' \} \).

We have the following two initial conditions:

\[ F^{c2v}(h, i', k) = 1; h = 0, k = 0, i = 0, 1, ..., \frac{L_c}{m}, \]
\[ i' = 0, ..., \min \{ h, i' \} \]

(49)

\[ F^{c2v}(h, i', k) = 0; H \geq h \geq 0, i = 0, 1, ..., \frac{L_c}{m}, \]
\[ i' = 0, ..., \min \{ h, i' \}; k \neq 0 \]

(50)

It can be seen that we can derive \( F(h, i', k) \) by solving the following \( H \) equations:

\[ F^{c2v}(0, i', k) = 0 \]
\[ F^{c2v}(1, i', k) = U^{c2v}_i \times F^{c2v}(0, i' - 1, k - 1) \]
\[ + (1 - U^{c2v}_i) \times (1 - F^{c2v}(0, i', k)) \]
\[ F^{c2v}(2, i', k) = U^{c2v}_i \times F^{c2v}(1, i' - 1, k - 1) \]
\[ + (1 - U^{c2v}_i) \times F^{c2v}(1, i', k) \]
\[ F^{c2v}(3, i', k) = U^{c2v}_i \times F^{c2v}(2, i' - 1, k - 1) \]
\[ + (1 - U^{c2v}_i) \times F^{c2v}(2, i', k) \]
\[ \cdots \cdots \cdots \]
\[ F^{c2v}(H - 1, i', k) = U^{c2v}_i \times F^{c2v}(H - 2, i' - 1, k - 1) \]
\[ + (1 - U^{c2v}_i) \times F^{c2v}(H - 2, i', k) \]
\[ F^{c2v}(H, i', k) = U^{c2v}_i \times F^{c2v}(H - 1, i' - 1, k - 1) \]
\[ + (1 - U^{c2v}_i) \times F^{c2v}(H - 1, i', k) \]

(51)

Next, we define \( K^{c2v}_{i'} \), as the average number of requests that a randomly picked peer, which has \( i \) useful pieces and is looking for \( \frac{L_c}{m} - i \) pieces in a specific time slot, will send to its neighbors. It can be derived by calculating the expected value of \( F(h, i', k) \), as below:
Expected - value$[F_{e^{2v}}(h, i', k)]$ = 

\[
\sum_{k=0}^{m} k \times F_{e^{2v}}(h, i', k)
\]

(52)

where $i = 0, 1, \ldots, \frac{L_c}{m}$ and $i' = 0, \ldots, \frac{L_c}{m} - i$

and $k = 0, \ldots, \min\{H, i'\}$

At any given time slot, we define $\overline{K}_{e^{2v}}$, as the average number of requests that any randomly picked peer sends to its neighbors at any given time slot. $K_{e^{2v}}$ can be expressed as:

\[
\overline{K}_{e^{2v}} = \sum_{i' = 0}^{L} (P_{e^{2v}} \times F_{e^{2v}})
\]

(53)

We also define $X_{e^{2v}}$, as the average number of requests that a randomly selected peer (e.g., peer A) receives from its neighbors in addition to the request received from peer D, as below:

\[
X_{e^{2v}} = \frac{(H - 1) \times \overline{K}_{e^{2v}}}{H}
\]

(54)

Now, we introduce $Q_{e^{2v}}$, as the probability that a randomly selected peer (e.g., peer A) fulfills a specific request (e.g., request from peer D) among all the requests that it received. $Q_{e^{2v}}$ can be expressed as follows:

\[
Q_{e^{2v}} = \frac{1}{1 + X_{e^{2v}}}
\]

(55)

where $i = 0, 1, \ldots, \frac{L_c}{m}$ and $i' = 0, \ldots, (\frac{L_c}{m} - i)$

Next, we define $r_{e^{2v}}^{i,n}$, as the probability that a peer, which has $i$ useful pieces at a given time slot, downloads $n$ pieces in the same time slot. We recall that, the maximum number of requests that can be sent by this peer will be $s_{e^{2v}}^{max} = \min\{\frac{L_c}{m} - i, H\}$. It is also obvious that the number of pieces which can be downloaded by this peer at any given time slot is a Binomial random variable with parameters $k$ and $Q_{e^{2v}}$ expressed as follows:

\[
r_{e^{2v}}^{i,n} = \min\{\frac{L_c}{m} - i, H\}
\]

\[
= \sum_{k=n}^{\min\{H, s_{e^{2v}}^{max}\}} \binom{k}{n} F_{e^{2v}}(h, i', k) \left(\frac{1}{H}\right)^k \left(1 - \frac{1}{H}\right)^{(H-k)}
\]

(56)

where $i = 0, 1, \ldots, \frac{L_c}{m}$ and $\sum_{i=0}^{m} (P_{e^{2v}}) = 1$

(57)

We assumed that the system is in the steady state, hence, the peer distribution does not change with time. The following is true in this situation:

In Figure 20 if we consider any particular state (e.g., $i$th state) and all the incoming and outgoing transitions to and from state $i$, we find the following relationship is true as shown in equation 58:

\[
(M_{i+1}^{e^{2v}} \times P_{i+1}^{e^{2v}}) + \sum_{k=0}^{\min\{H, i\}} \binom{L_c}{m} (r_{e^{2v}}^{i-k,k} \times P_{i-k}^{e^{2v}}) - \sum_{k=1}^{L_c-1} (r_{e^{2v}}^{i-k,k} \times P_{i}^{e^{2v}}) - (M_{i}^{e^{2v}} \times P_{e^{2v}}) = 0
\]

(58)

(59)

Fig. 20. A discrete-time Stochastic model for $C2V$ scenario

where $i = 0, 1, \ldots, \frac{L_c}{m}$ and $i' = 0, \ldots, (\frac{L_c}{m} - i)$, $s_{e^{2v}}^{max} = \min\{\frac{L_c}{m} - i, H\}$ and $n = 0, 1, \ldots, s_{e^{2v}}^{max}$

In our analysis, we assume that in a specific time slot, in a randomly picked peer’s buffer in the network, desired pieces, which are scheduled to be downloaded in this time slot, gets downloaded and stored in the buffer by that peer is uniformly distributed.

We define $M_{e^{2v}}$ as the probability that a randomly picked peer in the network with the buffer length of size $L_c$ and $i$ useful pieces among possible $(\frac{L_c}{m})$ sub-stream pieces, which are uniformly distributed, in its buffer and in a specific time slot, can play a piece in this time slot, would be expressed as follows:

\[
M_{e^{2v}} = \frac{i}{L_c}
\]

(57)
expressed as follows:

\[ P_{c_{2v}}^{cont} = \frac{L_c}{m} \sum_{i=0}^{\frac{L_c}{m}} P_{c_{2v}}^{i} \times M_{c_{2v}}^{i} \]  

(59)

We define \( d_{c_{2v}}^{i} \), as the average download rate of a peer that has \( i \) useful pieces in its buffer of size \( L_c \) in a specific time slot as follows:

\[ d_{c_{2v}}^{i} = \min \left\{ H, \frac{L_c}{m} - i \right\} \sum_{k=0}^{\frac{L_c}{m}} (k \times r_{c_{2v}}^{k}) \]  

(60)

where \( i = 0, 1, \ldots, \frac{L_c}{m} \).

In Figure 21 we see different states of a discrete-time Stochastic model for HnH multi-channel live streaming system.

V. RESULTS

VI. LIMITATIONS AND FUTURE WORK

We tried to propose models for three different scenarios of interactions among peers in a (proposed) HnH system and tried to capture key characteristics of that system. However, we made a lot of assumptions to make our model simpler. We mention some of them below:

- We have assumed unlimited download capacity and uniform upload capacity. In reality, most of the peers have heterogeneous upload capacity.
- We also assumed that the number of requests that any randomly selected peer in the network sends in any given time slot is a fixed number and based on that we have approximated the probability that a request of a randomly picked peer in the network, which has been sent to one of its neighbors, gets fulfilled.
- We have considered no or equal priority for fetching a piece in the playback buffer to make our analysis simpler. However, a higher priority should be given to the piece/pieces immediate to the playback pointer. We consider this to be our next work to be done.
- In this work, we have considered the three different scenarios where a peer picks up a particular type of neighbor (i.e., viewing peer or cooperating peer). It will be interesting to study the effect when a peer sends requests to any type of neighbor. We will consider this in our future work.

REFERENCES