Modeling and Performance Analysis of Peer-to-Peer Live Streaming Systems

Md. Istiaque Shahriar

A Thesis
In
The Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirement
For the Degree of
Doctor of Philosophy (Computer Science) at
Concordia University
Montréal, Québec, Canada

January 2019

Concordia University
School of Graduate Studies

This is to certify that the thesis prepared

By: **Md. Istiaque Shahriar**

Entitled: **Modeling and Performance Analysis of Peer-to-Peer Live Streaming Systems**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Computer Science)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________________ Chair
Dr. A. R. Sebak

______________________________ External Examiner
Dr. X. Lin

______________________________ Examiner
Dr. M. K. Mehmet Ali

______________________________ Examiner
Dr. Thomas Fevens

______________________________ Examiner
Dr. D. Goswami

______________________________ Supervisor
Dr. Brigitte Jaumard

______________________________ Supervisor
Dr. Dongyu Qiu

Approved ______________________________
Dr. Volker Haarslev, Graduate Program Director

January 08, 2019 ______________________________
Dr. Amir Asif, Dean
Gina Cody School of Engineering and Computer Science

Abstract

Modeling and Performance Analysis of Peer-to-Peer Live Streaming Systems

Md. Istiaque Shahriar, Ph.D.
Concordia University, 2019

In recent years, Internet has witnessed a rapid growth in P2P (peer-to-peer) applications, especially, in the live streaming domain. There have been several deployments of large-scale industrial level P2P live video systems, e.g., CoolStream, PPLive, Sopcast. 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 can host over 100 channels). It is expected that in near future, live streaming systems with hundreds of user-generated channels will likely have thousands of live channels in total. With such a large number of streaming channels and huge number of participants, there are still some challenging issues needed to be addressed for an efficient P2P live streaming system. This PhD thesis is organized around three such problems related to the P2P live streaming systems.

In the first research problem, our focus is on the Dedicated Channels used by a Small numbered Viewers (can be termed as DCSV channels for short) in a multi-channel live streaming system. Usually, these are user generated channels and they suffer adversely from poor channel performance, mainly, due to having a small number of participants. As a result, when a viewer of such a channel explicitly requests for a block of streaming content (commonly referred as a chunk), the probability that the chunk 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 have proposed HnH (short for Hand-in-Hand), a novel scheme of cross-channel resource sharing, in order to solve the performance problem of DCSV channels due to their small number of viewers. We next develop a discrete-time stochastic model in order to analyze its efficiency.

In the second research problem, we focus on the Free riders who only want to download and watch the streaming content from their neighboring peers but are unwilling to upload any content to their neighbors. The presence of free riders impose obstacle to the stability of any live streaming system because of consuming bandwidth from the system without significant contribution. We have investigated the performance of a Live streaming system with and

without the presence of a free riders. First, we develop a discrete-time stochastic model and then compare the probability of continuous playback without any free rider and with certain amount of free riders. Next, we introduce a simple incentive mechanism and modify our stochastic model in order to accommodate the incentive mechanism. Then we compare the result of probability of continuous playback with and without having an incentive mechanism. Our work shows that presence of an incentive mechanism improves the over all system performance.

In the third research problem, we focus on less motivated peers who are not interested to upload streaming contents to their neighbors if those contents are not from the channels they are watching. The context of this work is related to our first research problem where we have proposed the HnH scheme and for simplicity have considered that all the peers from all participating DCSV channels are motivated to cooperate. However, in practice, some of them may behave selfish and become less motivated to help peers from other channels. In this work, we investigate the performance of a HnH scheme based Live streaming system with and without the presence of less motivated peers. First, we develop a discrete-time stochastic model and then compare the probability of continuous playback without any selfish peer and with certain amount of selfish peers. Next, we introduce a simple incentive mechanism and modify our stochastic model accordingly. Finally, we compare the result of probability of continuous playback with and without incentive mechanism. Our work shows that presence of an incentive mechanism improves the over all system performance due to the fact that less motivated peers are motivated to cooperate more by the improved performance.

Contents

List of Figures

List of Tables

Acknowledgements

I wish to express my deep gratitude to my supervisors, Prof. Jaumard and Prof. Dongyu Qiu. Their encouragement, guidance, and deep support during my study enabled me to develop an understanding of the subject. Their wide knowledge, detailed and constructive comments have been a great value for me. Their timely financial support during my PhD helped me to concentrate on my work.

I would like to express my deep appreciation to the members of my doctoral committee for their valuable feedback on my thesis. Their comments, questions and suggestions have been very useful to improve my Ph.D. work.

I am grateful to all my student colleagues for their help and support.

I would like to thank my wife Rezwana, for her love, understanding, continuous support and encouragement. I would like to thank my children for their patience, support and believing in me.

Finally, I would like to thank my parents for their continuous moral support, affection and inspiration which have brought me here today.

Chapter 1

Introduction

1.1 Motivation

The term peer-to-peer, also commonly known as P2P, refers to a class of systems and applications that employ distributed resources to perform a function in a decentralized manner. With the persistent deployment of computers, P2P systems are receiving increasing attention in the research, product development, and investment communities. The P2P approach helps to improve system scalability by avoiding dependency on the centralized servers and eliminates the dollar cost of setting up costly client-server infrastructure by enabling direct communication among the participating clients. All the resources of the participants are eventually aggregated for the P2P system.

The era of today's P2P systems began with Napster [1] which launched a new application for sharing mp3 files over Internet in early 1999. The basis for a P2P architecture is consist of the collective resources of the individual peers instead of an underlying pre-established network architecture. That is why a P2P system is more attractive w.r.t. to scalability and flexibility compared to a traditional client-server system. Although the best way to achieve multi-cast over Internet is to use IP multi-cast, unfortunately, it is not much deployed due to several practical constraints. Among them, the need for infrastructure change and the lack of scalability (as all the routers on the way must maintain per multi-cast group state) are noteworthy. IP multi-cast can be very effective in dedicated infrastructures but could not gain acceptance in a larger scale. There is a variant of client-server architecture called Content Delivery Network (CDN). However, in a Content Delivery Network, there might be a bottleneck at the server side. This bottleneck eventually imposes a capacity limitation on the server side for upload bandwidth as well as number of supported simultaneous connections to the content server. Therefore, the CDN scheme is good for a fixed and predictable number of clients but not for a larger scale.

Because of all these constraints present in other schemes, the P2P architecture is gradually getting more attention and acceptance from the research community and the industry. Applications of P2P systems include file sharing, distributed computing, live streaming (i.e., mostly video streaming), IP telephony, etc. Due to wider range of usage and increased acceptance in the industry, new problems are being reported and challenging issues are emerging to the research community.

P2P technology over Internet started its popularity through the P2P file sharing systems. BitTorrent, Khaza, Azereus, uTorrent are some of the most popular file sharing applications. Generally, in a P2P file-sharing application, a user who wishes to share his/her digital media files, registers them using the local file sharing application. Once an interested user initiates his/her query about a file, several message exchanges take place among the users of the file sharing application and eventually it is identified which user is having this file. Some times more than one user can be identified as the owner of that file. Then, the file downloading starts with the respective protocol.

P2P live streaming applications gained their popularity after the file sharing applications. At present, there are many live streaming systems and most of them are capable of distributing hundreds of live channels, e.g., CoolStream [2], PPLive [3], Sopcast [4].

Live streaming is much more challenging than file sharing in a sense that it requires real-time response from the peers. In both mechanisms, a user has to download content from another user or peer.

In a file sharing application, the prime concern of a user is faster downloading (i.e., less downloading time) without any particular constraint. In a live streaming system, generally a user must finish downloading a piece of streaming content before the content starts to be played back otherwise there is no use of that content.

In addition to the delay constraint, user is also concerned about the playback quality of the streaming content, whereas in a file sharing system, playback quality is not an issue.

Generally, having a large number of participants, the peers of the popular channels experience satisfactory channel performance. This is, mainly, due to the increased likelihood of the availability of contents within the viewers of the channel when any of the participants explicitly requests for the contents. Again, as there is a large number of peers, many of them may possess high upload bandwidth. On the contrary, for the channels with smaller number of participants, the scenario is generally opposite. It has been observed that, in a normal situation, mostly the dedicated channels with small number of viewers face difficulties to maintain stable channel streaming rate [5]. From now on, we refer to these channels as DCSV channels (i.e., short for Dedicated Channels used by a Small-numbered Viewers).

The poor performance in a DCSV channel happens, mainly, due to the fact that it contains a small number of participants. As a consequence, when a viewer explicitly requests for a block of streaming content, the probability that a 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., in a popular channel). Moreover, a DCSV channel might not have peers with sufficient amount of spare upload capacity.

In order to resolve this issue, few techniques have been proposed. In some of those strategies, emphasis is given on cross channel resource sharing schemes. In those schemes, peers who are members of the popular channels and have sufficient amount of spare upload bandwidth are assigned as volunteers to upload to the DCSV channels.

It is obvious that the upload bandwidth capacity is an important factor for the successful operation of a channel in a P2P live streaming system. It is usually assumed that download capacity of any peer is unlimited, only the upload capacity is limited and costly. Normally, a peer will download from its neighboring peers and upload to them giving special preference to the peers from whom this peer has downloaded before. This is how the stability and harmony in a P2P network is naturally maintained.

However, there may be a group of selfish peers present in the system who only intend to download and view the content but do not want to share their upload bandwidth to re-distribute the contents. These selfish peers are generally called the Free riders.

In addition to these usual Free riders, there may be another group of selfish peers present in a cross-channel shared-resource system (e.g., in VUD [5]). These peers are less motivated and unwilling to upload streaming contents if those contents are not from the channels they are watching (i.e., they may choose to upload contents from the channel they are watching). Both types of selfish peers drain away the main resource (i.e., the aggregated upload capacity) of a P2P system. Proper functioning of a shared-resource P2P live streaming system will depend much on its efficiency to handle those two types of selfish peers. It is worth mentioning that the second category of selfish peers (i.e., less-motivated ones) are not mentioned yet in literature.

1.2 Problem statement

1.2.1 DCSV channels

In the first research problem, our focus is on the Dedicated Channels used by a Small number of Viewers (can be termed as DCSV channels for short) in a multi-channel live streaming system. Usually, these are 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 have proposed HnH (short for Hand-in-Hand), a novel scheme of cross-channel resource sharing, in order to solve the performance problem of DCSV channels due to their small number of viewers. We next develop a discrete-time stochastic model in order to analyze its efficiency.

1.2.2 Effect of Free riders

In our second research problem, we have focused on the Free riders who only want to download and watch the streaming content from their neighboring peers but are unwilling to upload any content to the neighbors. The presence of Free riders impose obstacle to the stability of any live streaming system because of consuming bandwidth from the system without significant contribution.

In this work, we investigate the performance of a Live streaming system with and without the presence of any Free riders. First, we have developed a discrete-time stochastic model and then compared the probability of continuous playback without any Free rider and with certain amount of Free riders. Next, we introduce a simple incentive mechanism and modify our stochastic model in order to accommodate the incentive mechanism. Then we compare the result of probability of continuous playback with and without having an incentive mechanism. Our work shows that presence of an incentive mechanism improves the over all system performance.

1.2.3 Effect of Less motivated Peers on HnH

In our third research problem, we have focused on less motivated peers who are not interested to upload streaming contents to their neighbors if those contents are not from the channels they are currently watching. In this case, we assume that we already have our proposed cross-channel resource sharing scheme, HnH, a novel scheme to solve the performance problem of DCSV channels due to their small number of viewers in place.

However, the success of such a cooperation scheme depends on the mutual cooperation among the participants. The presence of a less motivated neighbor from the helping channel affects the performance of the HnH scheme. In this work, we investigate the performance of a HnH scheme based P2P live streaming system when less motivated peers are present in the network.

First, we have developed a discrete-time stochastic model for the HnH based P2P live streaming system assuming selfish peers may be present in the system. Then we compare the probability of continuous playback assuming no selfish peers and then with certain amount of selfish peers.
Next, we introduce a simple incentive mechanism and made necessary modification in the stochastic model. Then we compare the result of probability of continuous playback with and without having an incentive mechanism. Our work shows that presence of an incentive mechanism improves the over all system performance with cooperation.

1.3 Literature

In this section we will survey the recent works on the performance analysis of small channels and performance analysis of channels having Free riders.

1.3.1 Small channel Performance

In the literature, there are few theoretical studies on the modeling and performance analysis of P2P systems. We first review the studies dealing with single-channel streaming systems. In one of the earliest work, Qiu *et al.* [6] developed a simple deterministic fluid model for mesh-based P2P file sharing systems which provides insights into the performance of a BitTorrent like network. The fluid model is described by a set of differential equations. Zhou *et al.* [7, 8] developed a simple probability model for data-driven systems to compare different chunk selection, downloading and peer selection strategies based on start up delay and continuity. They assumed independent (i.e., they do not know each other) and homogeneous peers in a symmetric network setting for their analytical model. Kumar *et al.* [9] provided a stochastic fluid model to explore the fundamental characteristics and limitations of swarm-based P2P streaming channels. They explored the effect of peer churn on a swarm-based channel where a large number of viewers are assumed to be either with or without a playback buffer. Massoulie *et al.* [10] studied the problem of efficient decentralized broadcasting in a P2P network with an homogeneous upload capacity and heterogeneous upload capacities, and proposed completely decentralized algorithms. Liu *et al.* [11] derived performance bounds for minimum server load, maximum streaming rate, and minimum tree depth under different peer selection constraints. Wu *et al.* [12] considered the problem of server capacity provisioning among multiple channels in order to maintain adequate streaming quality for each channel. Jin *et al.* [13]considered a game theory based approach to show the performance improvement when the peers are motivated to cooperate more. However, they have only considered single channel in their model. Kotevski *et al.* [14] developed a hybrid model for the performance analysis of P2P live video streaming systems. They evaluated performance of the system both with and without video buffer. They used queuing network and fluid model for video streaming. However, they focused mainly on the performance issues related to the single channel P2P live video streaming systems and they did not consider any inter-channel cooperation scenario. Zhang *et al.* [15] have developed an analytical model to classify and evaluate neighbors of a peer and chunk selection strategies for the push and pull based P2P streaming systems. Based on their findings, they propose a greedy selection mechanism where a peer will select a chunk nearest to its playback pointer. Chen *et al.* [16] have developed a fluid model for analyzing the performance of P2P live video streaming systems under flash crowd. However, they mainly focused on the quality of service and issues related to peer latency and system recovery time for single channel P2P live video streaming system under flash crowd. Qiu *et al.* [17] have developed a stochastic model for performance analysis of network coding based P2P live video streaming systems. They have provided results for different segment selection techniques and compared their performances.

Wu *et al.* [18] have proposed a framework and overlay design for multi-channel live video streaming which completely decouples the event of viewing a channel by a peer from the task of uploading the streaming content by the same peer. They have developed infinite-server queuing models for their system where every peer joining the system is requested to distribute streaming chunks to one or more streaming channels, where a peer is watching one channel while uploading to other channels. This puts a lot of overhead on the system to maintain many distribution groups together with their membership. Liang *et al.* [5] have proposed a partial decoupling strategy instead of a complete decoupling of the peer viewing and uploading operations in the context of multiple channels (not necessarily DCSV channels). Only some of the bandwidth-rich peers (and not all the peers) among all the channels will be assigned to a helping group and this group will be responsible for providing additional upload capacity to the peers lacking it. Unlike [18], they put lighter burden (e.g., maintaining single distribution group instead of many) on the system. However, the heavier burden is shifted on the helping peers who share their upload capacities with multiple peers from multiple channels. Liu *et al.* [19] developed a capacity model with and without node degree bound for a tree-based network. Therein, they worked on the effect of flash crowd on the requested network capacity in order to provide adequate services. Liu *et al.* [20] emphasized on content availability in the playback buffer of a peer in order to maximize the utilization of their upload capacity. In their model, they focused on minimum delay and proposed a snowball streaming algorithm for real time video streaming systems for tree-based models.

In Shahriar *et al.* [21], a generic stochastic model for peers cooperating with each other is presented where these cooperating peers are from two different channels with small number of viewers. In [22] the authors have presented another stochastic model which depicts interactions among the peers from two different channels with more realistic assumptions. However, none of these works explored the effect of cooperation on effective upload capacity. Some works in the literature focused on cross-channel resource sharing schemes (e.g., [18], [5]), however, none of the works provides adequate motivation for the peers to share their bandwidth with peers from a different channel. This is because, these sharing schemes involve uploading and downloading from peers who are watching different channels and do not suffer from poor performance. Due to absence of proper motivation, some of these peers may try to cheat the system in different ways in order to avoid the responsibility of helping peers from a different channel.

1.3.2 Free riders and Incentive

Free riders have been studied for long from different perspectives in different contexts. Free rider is first noted to be discussed by E.Ader *et al.* [23] in P2P context, in which the authors pointed out that there are many Free riders in the network who try to take advantage of the system. Several works have been published on Free riders problem in P2P File sharing systems where as there are few works done in the context of Live streaming system. Next, we present some of the works found in literature.

Free riders have been studied for File sharing systems, e.g., Bittorent network by Qiu *et al.* [24], and for single channel Live streaming systems as well, e.g., in Liu *et al.* [25]. Qiu *et al.* have shown that even if the tit-for-tat mechanism is very good to protect a network from Free riders in a File sharing system, the Bittorent protocol still has one option called 'optimistic unchoking' which is exploited by the Free riders for their benefit. In an optimistic unchoking a peer randomly chooses one of its uncooperative neighbor and uploads to it [26]. Usually, the uncooperative neighbor are treated as in chocked state and do not receive any upload from the peer. Qiu *et al.* have suggested a modification in the Bittorent protocol, more specifically in the optimistic unchoking technique and have shown this modified unchoking technique can prevent Free riders. Zhengye *et al.* [25] have suggested multiple layered coding as an incentive mechanism to avoid the tendency of Free riding. In this mechanism, based on the contribution of its upload bandwidth, the user will receive the viewing quality of the streaming video. Koo *et al.* [27] have suggested an incentive mechanism for a P2P Content distribution system, where the jobs may be divided into smaller units, and users have incentives to truthfully revealing willingness-to-pay for services. In this mechanism, the capacity that every user makes available to others in turn determines the amount of resources that user receives. However, this mechanism is designed for the P2P File sharing schemes. Habib *et al.* [28] have developed a score-based incentive mechanism for single channel media streaming. A peer which gets higher score is rewarded with more choice of neighbor selection and flexibilities. However, it does not address the issue of fulfilling their incentive mechanism honestly. Kumar *et al.* [29] have designed a pricing and allocation mechanism for a special case of P2P network that allows the users within a firm to effectively share their resources and avoid Free riding. Here, the optimal price of a task is determined based on the delay at each peer. Wu *et al.* [30] have developed an auction and bid based incentive mechanism for P2P File sharing systems in the context of social network using game theory. They also have suggested an extension of their model for real time Live streaming systems. Altman *et al.* [31] have developed a stochastic model to study behavior of the P2P networks those distribute non-authorized music, books, or articles. They even consider the presence of Free riders in that network. Then this model is used to predict the efficiency of the counter-measures taken by the content providers against those P2P networks. Zhao *et al.* [32] have developed an analytical model to study the behavior of the Video on Demand P2P network having Free riders in the system. Jin *et al.* [13] considered a game theoric approach to show the performance improvement when the peers in presence of the Free riders are motivated to cooperate more. However, they have only considered single channel scenario in their model. Shahriar *et al.* [21] have developed a generic stochastic model for peers cooperating with each other where these peers are from two different channels having small number of viewers. In [22] the authors have presented another stochastic model which depicts interactions among the peers from two different channels with greater details and more realistic assumptions. In [33] the authors have presented a new stochastic model which considers the presence of Free riders in live streaming systems. In this work, the authors show the impact of the Free riders on the playback performance of the peers of the system. However, their work does not discuss about the impact of incentive mechanism on the Free riders. Yeung *et al.* [34] have designed a tit-for-tat based incentive mechanism for single channel P2P media streaming and modeled this as two repeated game. In one of the repeated game, they have shown interactions between the streaming server and the immediate peers. The other one deals with the interaction between a peer and its neighbor. Although they have formulated an optimization model for the first case, they provided only the simulation results. However, none of the works mentioned above have analyzed the effect of Free riders with a stochastic model for Live streaming systems.

1.4 Thesis Contributions

The contributions of this thesis have been published in three papers. Here, we give a brief description of each of them below:
Chapter 2 [35]: In this work, we propose a new cross-channel cooperation scheme, HnH (short for Hand-in-Hand), among the peers from different DCSV channels where peers of a DCSV channel suffers from poor channel performance. We identified that one of the main reasons behind this poor performance is small number of participants of that channel. So, under the proposed HnH scheme, the number of effective participants of a small channel increases and eventually improves the channel viewing performance.
We developed a discrete-time stochastic model in order to analyze the efficiency of the HnH scheme. Our proposed HnH scheme relies on natural incentive for cooperation among the performance deprived peers of DCSV channels who are assumed to be naturally interested to help each other to achieve better performance.
Numerical experiments and simulations have been conducted in order to first validate the stochastic model, and then to evaluate the efficiency of the HnH scheme. Experiments show that the HnH scheme allows improvement in the quality of service so as to reach a satisfactory level for the viewers of the DCSV channels.
Chapter 3 [36]: The presence of Free riders affects the performance of peer-to-peer Live streaming systems. In this work, we investigate the performance of a Live streaming system with and without the presence of a Free riders. First, we develop a discrete-time stochastic model and then compare the probability of continuous playback without any Free rider and with certain amount of Free riders. Next, we introduce a simple incentive mechanism and modify our stochastic model in order to accommodate the incentive mechanism. Then we compare the result of probability of continuous playback with and without having an incentive mechanism. Our work shows that presence of an incentive mechanism improves the over all system performance.
Chapter 4 [37]: In Chapter 4, we work further on our previously proposed cross-channel resource sharing scheme, HnH. The success of such a cooperation scheme depends on the mutual cooperation among the participants. The presence of a less motivated neighbor from a helping channel affects the performance of the HnH scheme. In this work, we have focused on less motivated peers who are not interested to upload streaming contents to their neighbors if those contents are not from the channels they are currently watching. They are different from Free riders in sense that they do act like a Free rider within the channel they are watching. We investigate the performance of a HnH scheme based P2P live streaming

system when less motivated peers are present in the network. First, we develope a discrete-time stochastic model for the HnH based P2P live streaming system assuming selfish peers are present in the system. Then we compare the probability of continuous playback continuity assuming no selfish peers and then with certain amount of selfish peers. Next, we introduce a simple incentive mechanism and made necessary modification in the stochastic model. Then we compare the result of probability of continuous playback with and without having an incentive mechanism. Our work shows that presence of an incentive mechanism improves the over all system performance with cooperation. Chapter 5 Conclusion and Future work In Chapter 5, we conclude the thesis with thesis conclusion and future work.

Chapter 2

Modelling and Performance Analysis of HnH: A Novel Approach for Combining Live Streaming Channels with a Small Number of Viewers

2.1 Introduction

In recent years, the Internet community has observed a rise in P2P applications, especially, in the domain of live streaming systems. Many large-scale industrial P2P live video systems like CoolStream [2], PPLive [3], Sopcast [4] have been deployed during this era.

Several thousands of users can participate in these systems, simultaneously. Again, almost all of the live P2P video systems provide multiple channels (e.g., PPLive [3] is capable of hosting over hundreds of channels). Among these channels, a good number of them are user-generated and dedicated in nature. With the help of high-speed Internet and increasing availability of mobile data, live streaming systems are expected to have hundreds of dedicated and user-generated channels.

It is an usual practice among the live streaming systems is to organize the the participating peers of a channel into a mesh-based swarm like structure. In such a structure, the peers receive and distribute streaming contents among each other. Such a swarm formation works well in terms of quality of viewing when the number of viewing peers of a given channel is large enough. Such is the case for a popular channel. On the contrary, it has been observed (e.g., [18]) that user-generated and dedicated channels experience poor viewing performance, mainly, due to their smaller number of participants.

A viewer of a user-generated or a dedicated channel usually has a sufficient amount of upload capacity (e.g., upload capacity is higher or equal to the playback rate). However, such a viewer usually suffers from insufficient number of copies of streaming chunks. We perceive this issue, primarily, as a content-bottleneck problem and address it accordingly. Consequently, we propose HnH, a novel cross-channel resource-sharing scheme based on mutual cooperation among the set of peers which are viewers of the Dedicated Channels with a Small number of Viewers (in short DCSV) channels. Assuming a multi-channel system with an HnH scheme and its overlays are given, we next develop a stochastic model for evaluating different resource sharing scenarios. In an earlier work [21], we presented a simple stochastic model for peers, cooperating with each other from two different channels, where both the channels have small number of viewers. In [22], we presented another stochastic model considering more interaction options. Beside the proposed HnH system, the core contribution of this paper is the design of a discrete-time stochastic model in order to evaluate the performance of a HnH system in terms of fostering collaboration among peers of DCSV channels. The proposed model combines two or more live streaming channels such that some of the peers are viewing their respective channel whereas other peers are cooperating with the chunk forwarding of channels they are not viewing. We next derive nearly closed-form expressions for performance metrics such as the probability of continuity in streaming, consumption of effective bandwidth and the average number of downloaded streaming chunks per time slot. After the validation of the model, we use it to investigate the performance of a HnH scheme, as well as the influence of some of the network parameters: length of playback buffers, maximum delay from the source, effective upload capacity, on the performance. In the current work, we developed our own simulator and compared the simulation results with the numerical results. We also present an analysis and numerical results of the effect of cooperation on the upload capacity of a peer which is usually a concern for a cooperating peer.

This paper is organized as follows. In Section 2.2, we discuss the relevant recent work from the literature. In Section 2.3, we describe our proposed HnH multi-channel live streaming scheme. In Section 2.4, we next build a discrete-time stochastic model. In Section 2.5, we develop the computations of several probabilities out of the stochastic model of the previous section. We then use the proposed stochastic model and its derived probabilities to study the efficiency of the HnH model in the context of DCSV channels. Numerical results are presented and discussed In Section 2.6. Conclusions are derived in Section 2.7.

2.2 Related work

In the literature, there are few theoretical studies on the modeling and performance analysis of P2P systems. We first review the studies dealing with single-channel streaming systems. In one of the earliest work, Qiu *et al.* [6] developed a simple deterministic fluid model for mesh-based P2P file sharing systems which provides insights into the performance of a BitTorrent like network. The fluid model is described by a set of differential equations. Zhou *et al.* [7, 8] developed a simple probability model for data-driven systems to compare different chunk selection, downloading and peer selection strategies based on startup delay and continuity. They assumed independent (i.e., they do not know each other) and homogeneous peers in a symmetric network setting for their analytical model. Kumar *et al.* [9] provided a stochastic fluid model to explore the fundamental characteristics and limitations of swarm-based P2P streaming channels. They explored the effect of peer churn on a swarm-based channel where a large number of viewers are assumed to be either with or without a playback buffer. Massoulie *et al.* [10] studied the problem of efficient decentralized broadcasting in a P2P network with an homogeneous upload capacity and heterogeneous upload capacities, and proposed completely decentralized algorithms. Liu *et al.* [11] derived performance bounds for minimum server load, maximum streaming rate, and minimum tree depth under different peer selection constraints. Wu *et al.* [12] considered the problem of server capacity provisioning among multiple channels in order to maintain adequate streaming quality for each channel.

Jin *et al.* [13] considered a game theoric approach to show the performance improvement when the peers are motivated to cooperate more. However, they have only considered single channel in their model. Kotevski *et al.* [14] developed a hybrid model for the performance analysis of P2P live video streaming systems. They evaluated performance of the system both with and without video buffer. They used queuing network and fluid model for video streaming. However, they focused mainly on the performance issues related to the single channel P2P live video streaming systems and they did not consider any inter-channel co-operation scenario. Zhang *et al.* [15] have developed an analytical model to classify and evaluate neighbors of a peer and chunk selection strategies for the push and pull based P2P streaming systems. Based on their findings, they propose a greedy selection mechanism where a peer will select a chunk nearest to its playback pointer. Chen *et al.* [16] have developed a fluid model for analyzing the performance of P2P live video streaming systems under flash crowd. However, they mainly focused on the quality of service and issues related to peer latency and system recovery time for single channel P2P live video streaming system under flash crowd. Qiu *et al.* [17] have developed a stochastic model for performance analysis of network coding based P2P live video streaming systems. They have provided results for different segment selection techniques and compared their performances. Wu *et al.* [18] have proposed a framework and overlay design for multi-channel live video streaming which completely decouples the event of viewing a channel by a peer from the task of uploading the streaming content by the same peer. They have developed infinite-server queuing models for their system where every peer joining the system is requested to distribute streaming chunks to one or more streaming channels, where a peer is watching one channel while uploading to other channels. This puts a lot of overhead on the system to maintain many distribution groups together with their membership. Liang *et al.* [5] have proposed a partial decoupling strategy instead of a complete decoupling of the peer viewing and uploading operations in the context of multiple channels (not necessarily DCSV channels). Only some of the bandwidth-rich peers (and not all the peers) among all the channels will be assigned to a helping group and this group will be responsible for providing additional upload capacity to the peers lacking it. Unlike [18], they put lighter burden (e.g., maintaining single distribution group instead of many) on the system. However, the heavier burden is shifted on the helping peers who share their upload capacities with multiple peers from multiple channels. Liu *et al.* [19] developed a capacity model with and without node degree bound for a tree-based network. Therein, they worked on the effect of flash crowd on the requested network capacity in order to provide adequate services. Liu *et al.* [20] emphasized on content availability in the playback buffer of a peer in order to maximize the utilization of their upload capacity. In their model, they focused on minimum delay and proposed a snowball streaming algorithm for real time video streaming systems for tree-based models. In Shahriar *et al.* [21], a generic stochastic model for peers cooperating with each other is presented where these cooperating peers are from two different channels with small number of viewers. In [22] the authors have presented another stochastic model which depicts interactions among the peers from two different channels with more realistic assumptions. However, none of these works explored the effect of cooperation on effective upload capacity. Some works in the literature focused on cross-channel resource sharing schemes (e.g., [18], [5]), however, none of the works provides adequate motivation for the peers to share their bandwidth with peers from a different channel. This is because, these sharing schemes involve uploading and downloading from peers who are watching different channels and do not suffer from poor performance. Due to absence of proper motivation, some of these peers may try to cheat the system in different ways in order to avoid the responsibility of helping peers from a different channel.

In our proposed stochastic model, we have introduced cooperation among the peers from different DCSV channels where every peer is very likely to suffer from poor channel performance. We believe that this sufferings induce a sufficient motivation for all the peers to cooperate, even if it means downloading chunks from a channel, they are not interested in viewing. Especially, since this cooperation ultimately improves the performance of their own channel, this becomes a strong incentive for the peers across DCSV channels to help each other.

Therefore, the aim of our study is to go one step further in peer collaboration, and to evaluate the improvement of the performance resulting from such a uniform cooperation. It corresponds to a realistic approach of cross-channel resource sharing and cooperation among peers from different DCSV channels.

Some of the previous works have proposed cooperation schemes in different ways at different levels. They have proposed some models but mostly provided experimental results based on heuristics. Beside the proposed HnH cooperation scheme, the core contribution of this paper is the discrete-time stochastic model in order to evaluate the performance based on collaboration among peers of DCSV channels of a HnH system. We next derive nearly closed-form expressions for performance metrics such as the probability of continuity in streaming, consumption of effective bandwidth and the average number of downloaded streaming chunks per time slot. After the validation of the model, we use this model to investigate the performance of the HnH scheme, as well as the influence of some of the network parameters: length of playback buffers, maximum delay from the source, effective upload capacity, on the performance.

Note that, some previous works have focused on bandwidth bottleneck and solved the performance problem by adding additional upload capacity to those DCSV channels. However, additional capacity can be avoided if the performance issue is interpreted as a content bottleneck problem. The solution is then to define some mechanisms in order to increase the number of participants, which will eventually increase the number of instances of the DCSV chunks. In order to reduce the additional overhead cost, we consider cooperation at the sub-stream level.

2.3 HnH Scheme

We propose HnH, a multi-channel live streaming system which allows cooperation among performance-suffering peers of different DCSV channels. We consider that small number of participants is the key reason for poor viewing performance of the DCSV channels. More specifically, it is the absence of sufficient number of streaming chunks (rather than insufficient amount of upload capacity) that causes the problem. A possible solution to this problem could be to seek assistance from peers of a popular channel to inject streaming contents to a DCSV channel. However, these peers do not suffer from the unavailability of streaming chunks and hence have no sufficient motivation to help another (DCSV) channel.

On the contrary, the DCSV channel peers, who suffer from poor performance, can form combined channels in a HnH scheme based on multi-channels, because of their interest and mutual benefit. Once organized, all the peers viewing individual DCSV channels are considered as common members to all participating DCSV channels and each peer acquires a sufficient number of neighbors. Cooperating peers receive and distribute streaming contents of the participating channels in addition to its own viewing channel. Due to possible upload overhead on the cooperating peers in the combined channels, we propose cooperation at a sub-stream level. With the option of sub-streaming, chunks of a channel stream can be further divided into m number of sub-streams (i.e., subgroups) at the source server. For example, if the number of sub-streams is set to 5 (i.e., $m = 5$), all streaming chunks will be divided into 5 sub-streams at the server. In that case, sub-stream 1 corresponds to chunks $1, 6, 11, 16$ etc and sub-stream 2 corresponds to chunks $2, 7, 12, 17$ etc. Now, a viewing peer will always upload and download all the chunks of the stream (i.e all the sub-streams) among themselves. However, a cooperating peer may try to reduce its overhead of helping a viewing peer by only uploading and downloading the chunks corresponding to a particular sub-stream. Its cooperating buffer will contain fewer chunks than a viewing peer. In other words, instead of dealing with all the chunks of all the sub-streams, a cooperating peer may store and distribute only one of the m-sub streams of the viewing channel. In the HnH approach, a peer of a particular DCSV channel have two types of neighbors: *(i)* Viewing: peers who are watching the same channel (i.e., viewing channel) of that peer and *(ii)* Cooperating: peers from a different DCSV channel (i.e., a cooperating channel) who cooperate with a viewing peer in order to distribute its streaming chunks.

As illustrated in Figure 1, we consider the following interaction scenarios:

- **Viewing-to-Viewing (V2V) Scenario:** A viewing peer requests and downloads streaming contents to and from another viewing peer watching the same channel. As shown in Figure 1, the solid directed lines between viewing peer A and viewing peer B indicate V2V interactions. The Viewing-to-Viewing (V2V) scenario represents the interaction that usually takes place among the peers in a swarm-based P2P streaming system. However, two other interactions, Viewing-to-Cooperating (V2C) and Cooperating-to-Viewing (C2V), take place as part of our proposed HnH scheme.

- **Viewing-to-Cooperating (V2C) Scenario:** A viewing peer requests and downloads streaming contents from a cooperating peer of that channel. As shown in Figure 1,

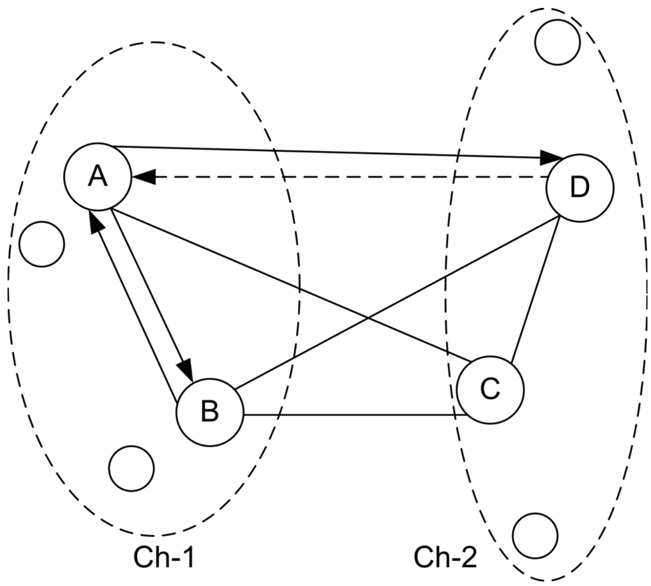


Figure 1: Peer interactions in the HnH scheme: an arrow from A to D means peer A requests and downloads streaming chunks from peer D

peer A is the viewing peer of channel Ch-1. Peer D is watching channel Ch-2 but working as a cooperating peer for channel Ch-1. The solid line from peer A to its cooperating neighbor D represents a V2C interaction. Viewing peer A requests and downloads from cooperating peer D similar to the V2V case. The only difference is that D may only buffer part of the stream (i.e., a sub-stream)

- **Cooperating-to-Viewing (C2V) Scenario:** A peer, which cooperates for a given channel, and that is not itself a viewer of this channel, requests and downloads streaming contents from a viewing peer. As shown in Figure 1, peer D is watching channel Ch-2 but working as a cooperating peer for channel Ch-1. Peer A is the viewing peer of channel Ch-1 and the dotted line from the cooperating neighbor D to the viewing peer A represents a C2V interaction. Cooperating peer D requests and downloads from viewing peer A in a way similar to the V2V case. However, the only difference is that D may only request and buffer part of the stream (i.e., a sub-stream) from its viewing neighbor A.

A possible interaction of the form of **Cooperating-to-Cooperating** (e.g., an interaction between two peers from channel Ch-2 for the purpose of getting a chunk of Ch-1) is not allowed in the proposed HnH scheme, as consideration of C2C interaction does not improve the performance but does make the model more complex.

2.4 Outline of the Stochastic Model

Assuming homogeneous (i.e., having identical parameters) viewing and cooperating peers from two channels, here we construct a stochastic model for the HnH scheme. Before discussing the details of the proposed stochastic model, we introduce the following notations.

2.4.1 Notations

We assume the peers to be homogeneous, i.e., to have identical parameters.

L_s	length of the viewing/playback buffer
L_c	length of the cooperating buffer, i.e., buffer of the cooperating peer for its non viewing channel
t_s	playback pointer at the source
$t_s - T_0$	earliest playback pointer at any peer
T	maximum delay between the playback pointer of the source and of a peer.
$t_s - T_1$	latest playback pointer at any peer
H_v	number of viewing neighbors of a peer
H_c	number of cooperating neighbors of a peer
H	total number of neighbors of a peer (i.e., $H = H_v + H_c$)
u_v	upload bandwidth that a viewing peer assigns for the viewing channel
u_c	upload bandwidth that a cooperating peer assigns for the cooperated channel
u	total upload bandwidth of a peer (i.e., $u = u_v + u_c$)
m	number of sub-streams at the source.

2.4.2 Assumptions

The model relies on the following assumptions:

- Each peer maintains a fixed number of neighbors, H. The download capacity of any peer is unlimited but the upload capacity is limited: u chunks/time slot (same for all peers).
- Each peer maintains a limited and fixed sized playback buffer, L_s, in its memory.
- While acting as a cooperating neighbor, a peer maintains a separate (cooperating) buffer, L_c. In order to reduce the overhead of helping a peer from another channel,

such a neighbor buffers and uploads only 1 out of m sub-streams (i.e., L_c/m chunks) of the viewing channel. However, it keeps track of all the L_c chunks of that viewing channel.

- A peer is said to have i useful (i.e., new) chunks (for itself) in its playback buffer L_s, if none of these i chunks have yet been played by this particular peer. These chunks cannot be overwritten by newer chunks before they are played by that peer.

- A peer's playback rate is 1 chunk/time slot and its playback pointer denotes the sequence number of the chunk the peer is playing at the current time slot.

- If a viewing peer contains j useful chunks in its buffer L_s then $L_s - j \geq 0$ old chunks may be also available in the buffer at the same time. Due to the limited size of the buffer, these $L_s - i$ old (i.e., already played) chunks can be overwritten by newer chunks.

- If a viewing peer with i useful chunks is looking for $L_s - i$ chunks, all the missing $L_s - i$ chunks have equal priority to be fetched. More specifically, no priority is given to chunks closer to the playback pointer.

- Upload bandwidth is set to u chunks/time slot, where $u \geq 1$ is a constant. However, download bandwidth is assumed to be unlimited since in most P2P systems, normally, download bandwidth is not a bottleneck issue.

2.4.3 Stochastic Model

The stochastic model (as illustrated in Figure 2) is described by $(L_s + 1)$ states (of a peer). Each state of the transition diagram corresponds to a peer state, as described by its number of useful chunks in the playback buffer. Associated with each state i (except initial one), there is exactly one reverse transition (M_i) to the its previous state (i.e., $i - 1$) which corresponds to the playback of one chunk/time slot. From each state i, there originates $L_s - i$ forward transitions, $r_{i,k}$, where $0 \leq k \leq L_s - i$. In the subsequent sections, we discuss transition probabilities in details.

The objective of the above Markov chain is to investigate the fundamental characteristics, limitations and performance problems of DCSV channels within a HnH system, depending on the values of the network parameters (e.g., buffer size, maximum delay, number of helping peers). After computing the transition probabilities, we will be able to estimate the probability distribution of the number of useful chunks. From these probability distributions, we calculate the probability of continuous playback to gain more information on how the

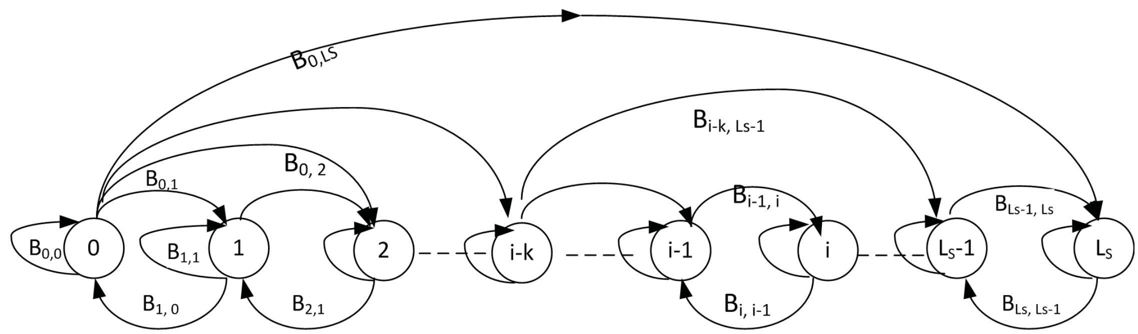


Figure 2: A discrete-time stochastic model for viewing peers in the HnH scheme having buffer size L_s.

continuous playback performance of such a HnH based system is controlled by the network parameters (e.g., buffer size, maximum delay, number of helping peers).

We have the following constraints:

- A streaming chunk, k, will be stored by peer A if it satisfies: $t_A \leq k \leq t_A + L_s - 1$ where, t_A is the playback pointer of A, otherwise, it will be discarded.
- A peer having i useful chunks in its buffer L_s sends a maximum of $S^{\max} = \min\{H, L_s - i\}$ requests to its H neighbors in the current time slot.
- The playback pointer t_A of peer A lies within the following limits: $t_s - T_1 \leq t_A \leq t_s - T_0$.

2.5 Streaming continuity Probabilities and Average Number of Downloaded Streaming Chunks

We now derive the probabilities of streaming continuity for the stochastic model (as illustrated in Figure 2), for each of the three scenarios identified in Section 2.4.

In a live streaming channel, the request of a peer to a neighbor in order to get a chunk, depends on the relative positions of their playback pointers and playback buffers We have the following two categories:

1. No overlapping of the chunks from the buffer of peer A with those from the buffer its neighbor, B.

2. Some overlapping of chunks from the buffer of peer A with those from the buffer of its neighbor, B.

Category 1 can be further sub-divided into two sub-categories:

1.1 Peer A is playing before neighbor B and there is no overlapping of chunks from their buffers (i.e., peer A has already played all the chunks that are currently present in the buffer of its neighbor B).

1.2 Neighbor B is playing way ahead of peer A such that even the old chunks of neighbor B are too recent for peer A to request (i.e., there is no overlapping of chunks).

and Category 2 into three sub-categories:

2.1 Peer A is playing ahead of neighbor B, however, there is some overlapping of useful chunks between buffers of A and B.

2.2 Neighbor B is playing ahead of peer A and there is some overlapping of useful and old chunks between buffers of A and B.

2.3 Neighbor B is playing way ahead of peer A however, there is some overlapping of old chunks of the buffer of B with the new chunks of the buffer of A.

These five general categories are illustrated in Figure 3. We next derive probabilities of the stochastic model for the V2V scenario.

2.5.1 Viewing-to-Viewing (V2V) Peers' Interactions

In the V2V scenario, consider an arbitrary peer (e.g., peer A) and one of its neighbors (e.g., peer B) such that both are of the same channel viewers with playback buffer of equal size, L_s. Suppose, at the current time slot, peer A has i useful chunks and neighbor B has j useful chunks in their respective buffers. t_A and t_B are the sequence numbers of the chunks to be played by the corresponding peer A and its neighbor B. Based on that, we calculate, U_i^{v2v}, the probability that peer A, while having i useful chunks in its buffer in a specific time slot, is interested in getting a chunk from a random neighbor, B. In order to calculate, U_i^{v2v}, let $f^{v2v}(i, j)$ be the probability that, at a given time slot, peer A with i useful chunks in its buffer is not interested in any of the j useful chunks of B. We then have:

$$U_i^{v2v} = \sum_{j=0}^{L_s} \left(1 - f^{v2v}(i, j)\right) P_j^v. \tag{1}$$

where P_j^v is the probability that a random viewing peer in the network has j useful chunks.

We calculate $f^{v2v}(i, j)$ for each of the following five mutually exclusive cases according to the relationship of the playback pointers, t_A and t_B:

$\text{Case}_1^{v2v} : t_s - T_1 \leq t_B \leq t_A - L_s$

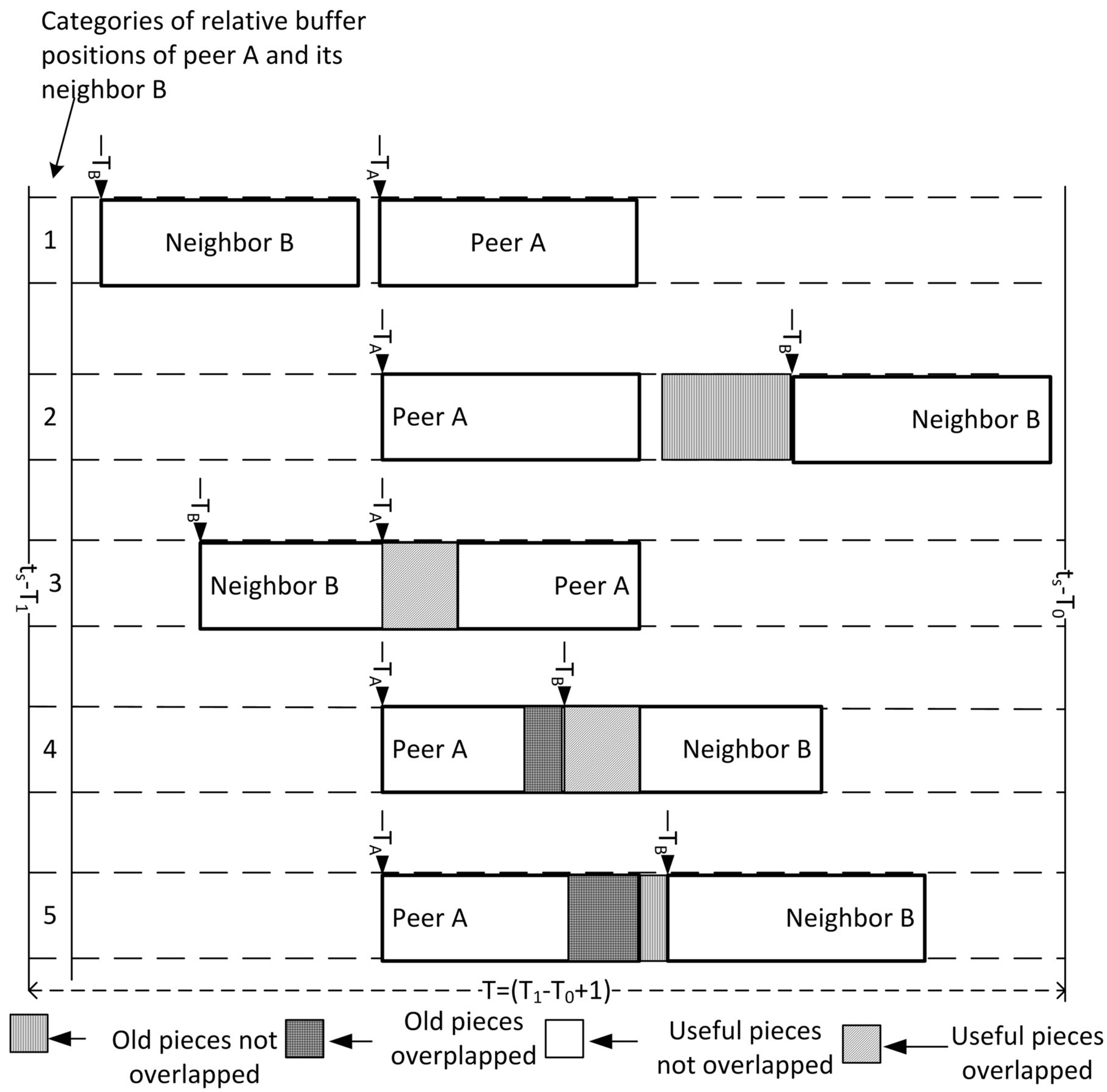


Figure 3: Relative positions of peer A and its neighbor B where t_A and t_B denote the playback pointer of the peer A and its neighbor B, respectively.

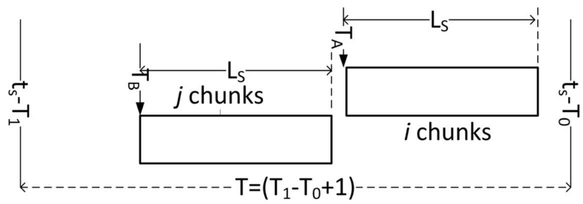
$\text{Case}_2^{v2v} : t_A + 2L_s - j \leq t_B \leq t_s - T_0$
$\text{Case}_3^{v2v} : t_A - L_s + 1 \leq t_B \leq t_A$
$\text{Case}_4^{v2v} : t_A + 1 \leq t_B \leq t_A + L_s - 1$
$\text{Case}_5^{v2v} : t_A + L_s \leq t_B \leq t_A + 2L_s - j - 1.$

We denote by $f_k^{v2v}(i, j)$ be the probability that peer A is not interested in the useful chunks of peer B and these two peers are in Case_k^{v2v}, where $k = 1...5$. We assume that the playback pointer of a peer is uniformly distributed within interval $[t_s - T_0, t_s - T_1]$ and $T = (T_1 - T_0 + 1)$. So, the probability that the playback pointer of peer A is at a given position is $1/T$. It is to be noted that, the definition of the probability of the playback pointer of any peer to be at a given position will be same for all the Case_k^{v2v} cases and the model will work with other distributions than the uniform one as well. However, it will be more complicated with other distributions.

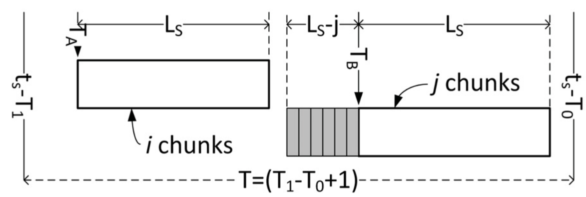
As the $f_k^{v2v}(i,j)$ probabilities are mutually exclusive cases:

$$f^{v2v}(i,j) \quad = \quad f_1^{v2v}(i,j) \;+\; f_2^{v2v}(i,j) \;+\; f_3^{v2v}(i,j) \;+\; f_4^{v2v}(i,j) \;+\; f_5^{v2v}(i,j). \tag{2}$$

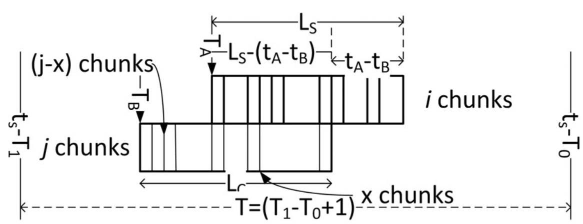
Computations of the $f_k^{v2v}(i,j)$ probabilities are as follows.



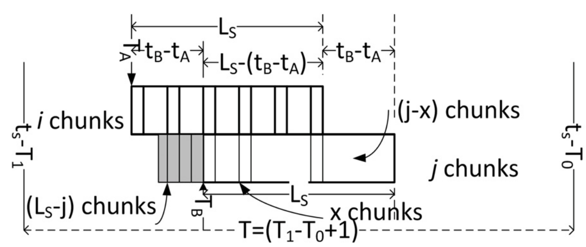
(a) $Case_1^{v2v}$



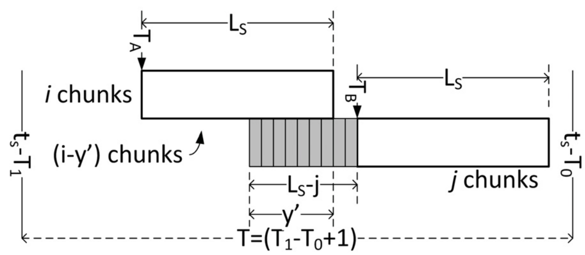
(b) $Case_2^{v2v}$



(c) $Case_3^{v2v}$



(d) $Case_4^{v2v}$



(e) $Case_5^{v2v}$

Figure 4: Overlapping of i useful streaming chunks of peer A with j useful chunks of one its random neighbor B in the Viewing-to-Viewing (V2V) cases

$Case_1^{v2v}$: $t_s - T_1 \leq t_B \leq t_A - L_s$ (see Figure 4(a)):

Peer A has i useful chunks and its random neighbor B has j useful chunks in their respective buffers. Moreover, those buffers have no content overlap. Peer A has already played all the j chunks of B and has no further interest in them. It follows:

$$f_1^{v2v}(i,j) = \sum_{t_A=t_s-T_1+L_s}^{t_s-T_0} \sum_{t_B=t_s-T_1}^{t_A-L_s} \frac{1}{T}.\frac{1}{T}. \tag{3}$$

In this case, probability that A is not interested in the chunks of $B = 1$. Here, probability that the playback pointer, t_A, of peer A is at a given position is $1/T$ and the probability that the playback pointer t_B, of peer B is at a given position is also $1/T$.
$Case_2^{v2v} : t_A + 2L_s - j \leq t_B \leq t_s - T_0$ (see Figure 4(b)):
In this case, buffers of peer A and neighbor B have no content overlap and as opposed to $Case_1^{v2v}$, buffer of the neighbor B contains more recent pieces than peer A. The lowest sequence number of the $(L_s - j)$ old chunks (residing in the buffer) of peer B is even higher than the highest sequence number of the chunks that A can store in its buffer in the current time slot. Hence, peer A has no interest in its neighbor B. It follows:

$$f_2^{v2v}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0-(2L_s-j)} \sum_{t_B=t_A+2L_s-j}^{t_s-T_0} \frac{1}{T^2}. \tag{4}$$

In this case, probability that A is not interested in the chunks of $B = 1$.
$Case_3^{v2v}[t_A - L_s + 1 \leq t_B \leq t_A]$ (see Figure 4(c)):
Among the j useful chunks of neighbor B, x chunks are already present in the overlapping portion of the buffer of peer A. If A is not interested in B, then whatever chunks neighbor B has in the overlapping portion, peer A already has those chunks. The remaining $(j - x)$ chunks of B are of no interest to A (as those chunks are already played by A). It follows:

$$f_3^{v2v}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_B=\max\{t_s-T_1,t_A-L_s+1\}}^{t_A} \sum_{x=\max\{0,j-(t_A-t_B)\}}^{\min\{L_s-(t_A-t_B),i,j\}} \left[\left(\frac{\binom{t_A-t_B}{j-x}\binom{L_s-(t_A-t_B)}{x}}{\binom{L_s}{j}}\right)\left(\frac{\binom{L_s-x}{i-x}}{\binom{L_s}{i}}\right)\right]\frac{1}{T^2}. \tag{5}$$

In this case, probability that A is not interested in the chunks of B is computed inside the square braces. The first part inside the square braces denotes the probability that neighbor B has j chunks in its buffer and among these j chunks, x chunks are in the portion (i.e., $L_s - (t_A - t_B)$) which is overlapped with the buffer of peer A. Remaining $j - x$ chunks of peer B lies in the non-overlapped region, $t_A - t_B$. The second part denotes that peer A has

i chunks where these i chunks include all the x chunks which are in the buffer of neighbor B.

$Case_4^{v2v}: t_A + 1 \leq t_B \leq t_A + L_s - 1$ (see Figure 4(d)):

Neighbor B has x (out of j) useful chunks in the overlapping portion. If peer A is not interested in B then A also has all the x chunks in its buffer. Moreover, peer A has all the $(L_s - j)$ old chunks of B. The remaining $(j - x)$ chunks of B are of no interest to A (as those chunks are too new for it for the current time slot). We then have:

$$f_4^{v2v}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_B=\min\{t_s-T_0,t_A+1\}}^{\min\{t_s-T_0,t_A+L_s-1\}} \sum_{x=\max\{0,j-(t_B-t_A)\}}^{\min\{L_s-(t_B-t_A),i-\min\{(t_B-t_A),(L_s-j)\},j\}} \left[\left(\frac{\binom{t_B-t_A}{j-x}\binom{L_s-(t_B-t_A)}{x}}{\binom{L_c}{j}} \right) \left(\frac{\binom{L_s-y}{i-y}}{\binom{L_s}{i}} \right) \right] \frac{1}{T^2}. \quad (6)$$

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

Here, probability that A is not interested in the chunks of B is computed inside the square braces where the first part inside the square braces denotes the probability that neighbor B has j chunks in its buffer and among these j chunks, x chunks are in the portion, $L_s-(t_B-t_A)$, which is overlapped with the buffer of peer A. Remaining $j - x$ useful chunks lies in the non-overlapped region, $t_B - t_A$, of peer B. However, in this case, B has $L_s - j$ old chunks which are overlapped with the first $t_B - t_A$ portion of the buffer of peer A The second part denotes that peer A has i chunks where these i chunks include all the x useful chunks and all the $L_s - j$ old chunks of the buffer of neighbor B (i.e., defined under y).

$Case_5^{v2v}: t_A + L_s \leq t_B \leq t_A + 2L_s - j - 1]$(see Figure 4(e)):

In this case, there is a overlapping between the i useful chunks of peer A and y' (among $(L_s - j)$) old chunks of neighbor B. Peer A is not interested in the j useful chunks of neighbor B (as those are too new for A in the current time slot). If A is not interested in B then A already have all the y' old chunks within its i chunks. We have:

$$f_5^{v2v}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0-L_s} \sum_{t_B=t_A+L_s}^{\min\{t_s-T_0,[(t_A+L_s)+L_s-j-1]\}} \left[\frac{\binom{L_s-y'}{i-y'}}{\binom{L_s}{i}} \right] \frac{1}{T^2}. \quad (7)$$

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

In this case, probability that A is not interested in the chunks of B is computed inside the square braces. It denotes the probability that neighbor B has $(L_s - j)$ old chunks in its buffer and among these $(L_s - j)$ chunks, y' chunks are in the portion which is overlapped with the buffer of peer A. Peer A has i chunks where these i chunks include all the old y'

chunks which are in the buffer of neighbor B.

2.5.2 Viewing-to-Cooperating (V2C) Peers' Interactions

In the V2C scenario, consider an arbitrary peer (e.g., peer A with playback buffer L_s) and one of its cooperating neighbors (e.g., peer D with playback buffer L_c) such that they are viewers of two different channels. Suppose, at the current time slot, peer A has i useful chunks and neighbor D has j useful chunks in their respective buffers. For simplicity, we assume $L_s = L_c$. The streaming content is divided into m sub-streams and neighbor D stores only one of the m sub-streams (i.e., L_c/m chunks) in its buffer to extend its cooperation. Based on that, we will calculate, U_i^{v2c}, probability that peer A having i useful chunks in its buffer in a specific time slot, is interested to get a chunk from a random cooperative neighbor D. In order to calculate, U_i^{v2c}, let $f^{v2c}(i,j)$ be the probability that, at a given time slot, peer A with i useful chunks in its buffer is not interested in any of the j useful chunks of D. We then have:

$$U_i^{v2c} = \sum_{j=0}^{L_c/m} \left(1 - f^{v2c}(i,j)\right) P_j^c. \tag{8}$$

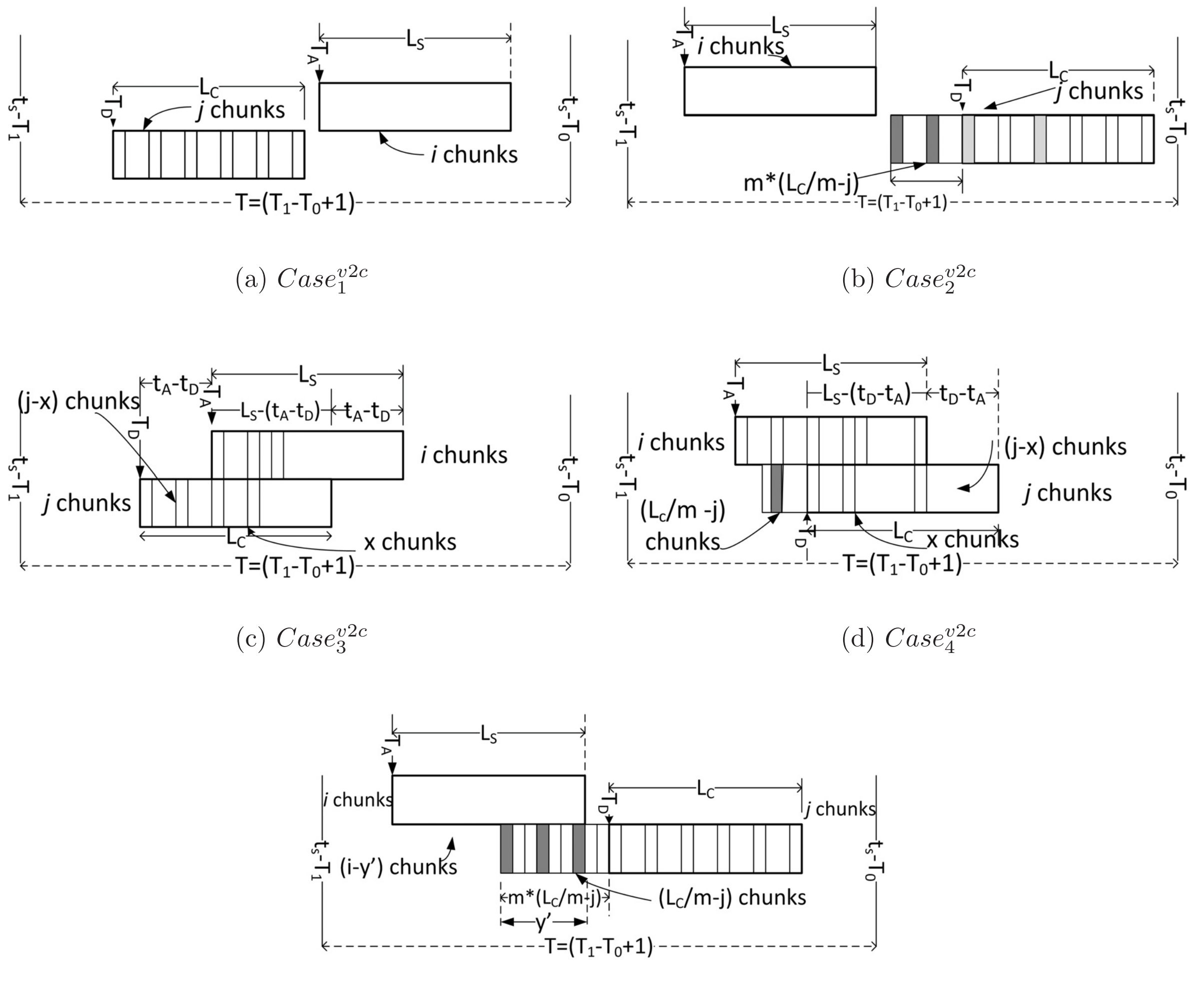
where P_j^c is the probability that a random cooperating peer in the network has j useful chunks.

We calculate $f^{v2c}(i,j)$ for each of the following five mutually exclusive cases according to the relationship of the playback pointers, t_A and t_D:

$\text{Case}_1^{v2c} : t_s - T_1 \le t_D \le t_A - L_c$
$\text{Case}_2^{v2c} : t_A + L_s + m(L_c/m - j) \le t_D \le t_s - T_0$
$\text{Case}_3^{v2c} : t_A - L_c + 1 \le t_D \le t_A$
$\text{Case}_4^{v2c} : t_A + 1 \le t_D \le t_A + L_s - 1$
$\text{Case}_5^{v2c} : t_A + L_s \le t_D \le t_A + L_s + m(L_c/m - j) - 1$
We denote by $f_k^{v2c}(i,j)$ the $f^{v2c}(i,j)$ probability in Case_k^{v2c}, where $k = 1...5$. As the $f_k^{v2c}(i,j)$ probabilities are mutually exclusive:

$$f^{v2c}(i,j) = f_1^{v2c}(i,j) + f_2^{v2c}(i,j) + f_3^{v2c}(i,j) + f_4^{v2c}(i,j) + f_5^{v2c}(i,j). \tag{9}$$

Computations of the $f_k^{v2c}(i,j)$ probabilities are as follows. We, now calculate the probability, that peer A will not be interested in the chunks of cooperating neighbor D.
Similar to Section 2.5.1, we assume that the playback pointer of a peer is uniformly distributed within the interval $[t_s - T_0, t_s - T_1]$ and $T = (T_1 - T_0 + 1)$.



(a) $Case_1^{v2c}$

(b) $Case_2^{v2c}$

(c) $Case_3^{v2c}$

(d) $Case_4^{v2c}$

(e) $Case_5^{v2c}$

Figure 5: Overlapping streaming chunks of peer A and one of its cooperative neighbor D, in the Viewing-to-Cooperative (V2C) cases

$Case_1^{v2c}$ $[t_s - T_1 \leq t_D \leq t_A - L_c]$ (see Figure 5(a)):
Peer A has i useful chunks in its buffer, L_s and cooperating neighbor D has j useful chunks in its buffer, L_c. The relative position of their playback pointers are such that there is no overlapping of contents between the buffers of these two peers. Peer A has already played back all the j chunks residing in the buffer L_c of neighbor D and has no further interest in them.

$$f_1^{v2c}(i,j) = \sum_{t_A=t_s-T_1+L_c}^{t_s-T_0} \sum_{t_D=t_s-T_1}^{t_A-L_c} \frac{1}{T^2}. \qquad (10)$$

$Case_2^{v2c}[t_A + L_s + m(L_c/m) - j \leq t_D \leq t_s - T_0]$ (see Figure 5(b)):
In this case, there is no overlapping of the two buffers of peer A and the cooperating neighbor

D. As opposed to $Case_1^{v2c}$, buffer of the neighbor D contains more recent pieces than peer A. The lowest sequence number of the $(L_c/m - j)$ old chunks of peer D is even higher than the highest sequence number of the chunks that A can store in its buffer in the current time slot.

$$f_2^{v2c}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0-(L_s+m((L_c/m)-j))} \sum_{t_D=t_A+L_s+m((L_c/m)-j)}^{t_s-T_0} \frac{1}{T^2}. \tag{11}$$

$Case_3^{v2c}[t_A - L_c + 1 \le t_D \le t_A]$ (see Figure 5(c)):
Viewing peer A has i useful chunks and cooperating neighbor D has j useful chunks in their corresponding buffers. Among the j chunks of cooperative neighbor D, x are located in the overlapping portion of the buffer (with peer A). If A is not interested in D then, whatever chunks (i.e., x) peer D has in the overlapping portion, peer A also has those x chunks in its buffer. The remaining $(j-x)$ chunks are of no interest to A (as playback pointer of A has already passed those chunks).

$$f_3^{v2c}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_D=\max\{t_s-T_1,t_A-L_c+1\}}^{t_A} \sum_{x=\max\{0,i-(t_A-t_D)\}}^{\min\{L_c-(t_A-t_D),i,j\}} \left[\left(\frac{\binom{(t_A-t_D)/m}{j-x}\binom{(L_c-(t_A-t_D))/m}{x}}{\binom{L_c/m}{j}}\right)\left(\frac{\binom{L_s-x}{i-x}}{\binom{L_s}{i}}\right)\right]\frac{1}{T^2}. \tag{12}$$

$Case_4^{v2c}[t_A + 1 \le t_D \le t_A + L_s - 1]$ (see Figure 5(d)):
Cooperative peer D has j useful chunks and $(L_c/m - j)$ old chunks in its buffer, L_c. Among the j useful chunks, D has x chunks in the overlapping buffer and A has all these x. Moreover, peer A has all the $(L_c/m - j)$ old chunks of D. The remaining $(j-x)$ chunks are of no interest to A (as those chunks are too new for A in the current time slot).

$$f_4^{v2c}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T-0-1} \sum_{t_D=t_A+1}^{\min\{t_s-T_0,[(t_A+L_s)-1]\}} \sum_{x=\min\{0,(i-(t_D-t_A)/m)\}}^{\min\{L_s-(t_D-t_A),i-\min\{(t_D-t_A),L_c/m-j\},j\}} \left[\left(\frac{\binom{(t_D-t_A)/m}{j-x}\binom{(L_c-(t_D-t_A))/m}{x}}{\binom{L_c/m}{j}}\right)\left(\frac{\binom{L_s-y}{i-y}}{\binom{L_s}{i}}\right)\right]\frac{1}{T^2}. \tag{13}$$

where $y = x + \min\{(t_D - t_A)/m, (L_c/m - j)\}$
$Case_5^{v2c}[t_A + L_s \le t_D \le t_A + L_s + m(L_c/m - j) - 1]$ (see Figure 5(c)):
In this case, there is no overlapping among the useful chunks of peer A and neighbor D. However, the i useful chunks of peer A has overlapping with y'/m (out of $m(L_c/m-j)$) old chunks of D and A has all the y'/m chunks in its buffer. Peer A will not be interested about

the j chunks of neighbor D (as those chunks are too new for the current time slot).

$$f_5^{v2c}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0-L_s} \sum_{t_D=t_A+L_s}^{(t_A+L_s)+m(L_c/m-j)-1} \left[\frac{\binom{L_s-y'}{i-y'}}{\binom{L_s}{i}}\right] \frac{1}{T^2}. \quad (14)$$

where $y' = t_A + L_s - [t_D - m(L_c/m - j)]$.

2.5.3 Cooperating-to-Viewing (C2V) Peers' Interactions

In the C2V scenario, consider an arbitrary cooperating peer (e.g., peer D with playback buffer L_c) and one of its viewing neighbors (e.g., peer A with playback buffer L_S) such that they are viewers of two different channels. Suppose, at the current time slot, peer D has i useful chunks and its neighbor A has j useful chunks in their respective buffers. For simplicity, we assume $L_c = L_s$. The streaming content is divided into m sub-streams and peer D stores only one of the m sub-streams (i.e., L_c/m chunks) in its buffer to extend its cooperation. Based on that, we will calculate, U_i^{c2v}, probability that peer D having i useful chunks in its buffer in a specific time slot, is interested to get a chunk from a random viewing neighbor A. In order to calculate, U_i^{c2v}, let $f^{c2v}(i,j)$ be the probability that, at a given time slot, peer D with i useful chunks in its buffer is not interested in any of the j useful chunks of viewing neighbor A. We then have:

$$U_i^{c2v} = \sum_{j=0}^{L_s} \left(1 - f^{c2v}(i,j)\right) P_j^v. \quad (15)$$

where P_j^{c2v} is the probability that a random viewing peer in the network has j useful chunks. Computations of the $f_k^{c2v}(i,j)$ probabilities are done as follows according to the relationship of the playback pointers, t_A and t_D:

$Case_1^{c2v} : t_s - T_1 \le t_A \le t_D - L_s$
$Case_2^{c2v} : t_D + L_c + L_s - j \le t_A \le t_s - T_0$
$Case_3^{c2v} : t_D - L_c + 1 \le t_A \le t_D$
$Case_4^{c2v} : t_D + 1 \le t_A \le t_D + L_c - 1$
$Case_5^{c2v} : t_D + L_c \le t_A \le t_D + L_c + L_s - j - 1$

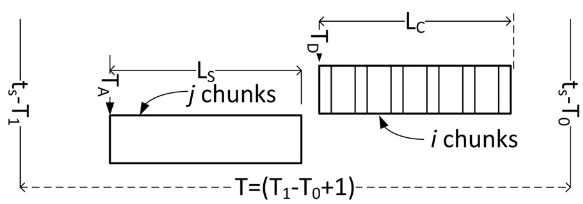
We denote by $f_k^{c2v}(i,j)$ the $f^{c2v}(i,j)$ probability in Case_k^{c2v}, where $k = 1...5$. As the $f_k^{c2v}(i,j)$

probabilities are mutually exclusive:

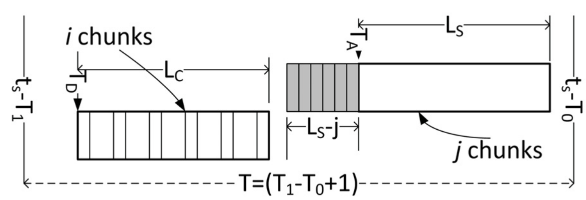
$$f^{c2v}(i,j) \quad = \quad f_1^{c2v}(i,j) \;+\; f_2^{c2v}(i,j) \;+\; f_3^{c2v}(i,j) \;+\; f_4^{c2v}(i,j) \;+\; f_5^{c2v}(i,j). \quad (16)$$

We, now calculate the probability, that peer D will not be interested in the chunks of its viewing neighbor A. Similar to Section 2.5.1, we assume that the playback pointer of a peer is uniformly distributed within the interval $[t_s - T_0, t_s - T_1]$ and $T = (T_1 - T_0 + 1)$. As compared to the previous two scenarios (i.e., V2V and V2C), C2V cases are more restricted in the sense that peer D is interested in those chunks (from its neighbor A) which correspond only to the particular sub-stream that peer D buffers at the current time slot.

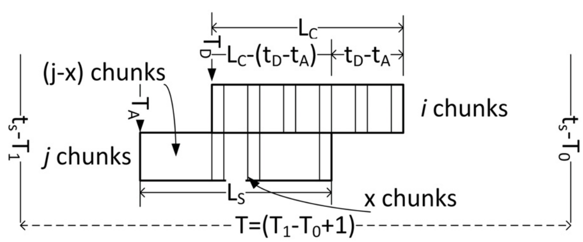
$Case_1^{c2v}[t_s - T_1 \leq t_A \leq t_D - L_s]$ (see Figure 6(a)):



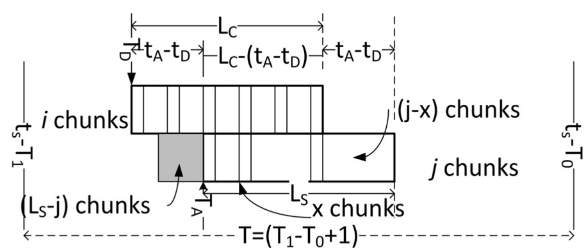
(a) $Case_1^{c2v}$



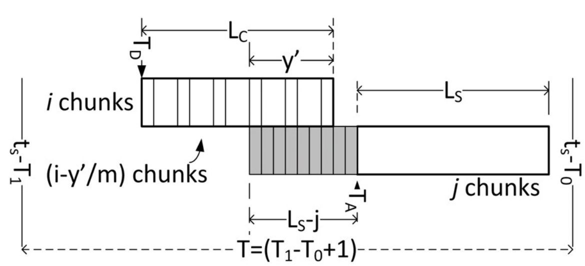
(b) $Case_2^{c2v}$



(c) $Case_3^{c2v}$



(d) $Case_4^{c2v}$



(e) $Case_5^{c2v}$

Figure 6: Overlapping streaming chunks of cooperating peer D and one of its viewing neighbor A, in the Cooperative-to-Viewing (C2V) cases

There is no overlapping of the respective buffers of viewing neighbor A and cooperating peer D. The playback pointer of peer D is already ahead of the playback pointer of neighbor A. It is obvious that peer D has no interest in the chunks of A irrespective of D's specific sub-stream.

$$f_1^{c2v}(i,j) = \sum_{t_D=t_s-T_1+L_s}^{t_s-T_0} \sum_{t_A=t_s-T_1}^{t_D-L_s} \frac{1}{T^2}. \tag{17}$$

$Case_2^{c2v}[t_D + L_c + L_s - j \leq t_A \leq t_s - T_0]$ (see Figure 6(b)):
In this case, the playback pointer of the cooperating peer D plays behind the playback pointer of its viewing neighbor A and there is no overlapping of their respective buffers. As opposed to $Case_1^{c2v}$, buffer of the neighbor A contains more recent pieces than peer D. The lowest sequence number of the $(L_s - j)$ old chunks (in the buffer) of A is even higher than the highest sequence number of the chunks that D can store in its buffer in the current time slot. Hence, peer D, irrespective of its specific sub-stream, has no interest in the chunks of its neighbor A.

$$f_2^{c2v}(i,j) = \sum_{t_D=t_s-T_1}^{t_s-T_0-(L_s+L_c-j)} \sum_{t_A=t_D+L_s+L_c-j}^{t_s-T_0} \frac{1}{T^2}. \tag{18}$$

$Case_3^{c2v}[t_D - L_s + 1 \leq t_A \leq t_D]$ (see Figure 6(c)):
Among the j useful chunks of neighbor A, x chunks are already present in the overlapping portion of the buffer of peer D. In other words, corresponding to the particular sub-stream of D, whatever chunks neighbor A has in the overlapping portion, peer D also has those chunks in the same portion. The remaining $(j - x)$ chunks of A are of no interest to D (as those chunks are already played by D).

$$f_3^{c2v}(i,j) = \sum_{t_D=t_s-T_1}^{t_s-T_0} \sum_{t_A=\max\{t_s-T_1,t_D-L_s+1\}}^{t_D} \sum_{x=\max\{0,i-(t_D-t_A)/m\}}^{\min\{(L_s-(t_D-t_A))/m,i,j\}} \left[\left(\frac{\binom{(t_D-t_A)/m}{j-x}\binom{L_s-(t_D-t_A)/m}{x}}{\binom{L_s}{j}} \right) \left(\frac{\binom{(L_c/m)-x}{i-x}}{\binom{L_c/m}{i}} \right) \right] \frac{1}{T^2}. \tag{19}$$

$Case_4^{c2v}[t_D + 1 \leq t_A \leq t_D + L_c - 1]$ (see Figure 6(d)):
Among the j useful chunks, neighbor A has x chunks in the overlapping portion of its buffer and peer D already have all those x chunks. Moreover, peer D has all the $(L_s/m - j)$ old chunks of neighbor A corresponding to the sub-stream of D. Peer D is not interested in the

remaining $(j-x)$ chunks of A as those chunks are too new for the current time slot.

$$f_4^{c2v}(i,j) = \sum_{t_D=t_s-T_1}^{t_s-T_0} \sum_{t_A=\min\{t_s-T_0,t_D+1\}}^{\min\{t_s-T_0,(t_D+L_c-1)\}} \sum_{x=\max\{0,i-\min\{i,((t_A-t_D)/m)\}\}}^{\min\{L_s-(t_A-t_D),i-\min\{(t_A-t_D)/m,(L_c/m-j)\}\}} \left[\left(\frac{\binom{(t_A-t_D)/m}{j-x}\binom{L_s-(t_A-t_D)/m}{x}}{\binom{L_s}{j}}\right)\left(\frac{\binom{(L_c/m)-y}{i-y}}{\binom{L_c/m}{i}}\right)\right]\frac{1}{T^2}. \quad (20)$$

where $y = x + (1/m)\min\{[(t_A - t_D), (L_s - j)]\}$

$Case_5^{c2v}[t_D + L_c \leq t_A \leq t_D + L_c + L_s - j - 1]$ (see Figure 6(e)):
There is no overlapping of useful chunks of the respective buffers of D and A. However, useful chunks of peer D has overlapping with y' old chunks of peer A and peer D has all those y'/m chunks in its buffer. It is obvious that peer D will not be interested about the j useful chunks of peer A as those chunks are too new for peer D to buffer in the current time slot.

$$f_5^{c2v}(i,j) = \sum_{t_D=t_s-T_1}^{t_s-T_0-L_s} \sum_{t_A=t_D+L_c}^{\min\{t_s-T_0,[(t_D+L_c)+L_s-j-1]\}} \left[\frac{\binom{(L_c/m)-y'/m}{i-y'/m}}{\binom{L_c/m}{i}}\right]\frac{1}{T^2}. \quad (21)$$

where $y' = t_D + L_c - [t_A - (L_s - j)]$.

Next, we study the requesting and downloading of chunks in the $V2V$, $V2C$ and $C2V$ cases. Since a cooperating peer download only from its viewing neighbors alone (as we do not consider the $C2C$ case), we begin with the computation for the $C2V$ case for its nature of simplicity.

2.5.4 Probabilities of Requesting and Downloading Chunks for a Cooperating Peer

We recall that, any random cooperating peer in this network has H_v fixed number of viewing neighbors and a buffer L_c of fixed length. It is also assumed that at any time slot if any randomly picked viewing peer (e.g. peer A) receives more than one request, u of them (including the one from D) will be fulfilled randomly by peer A.

We define $s_{\max}^{c2v}$, the maximum number of requests that an arbitrary cooperating peer (e.g., D) having i chunks in the current time slot can send to its neighbors as:
$s_{\max}^{c2v} = \min\{H_v, (L_c/m - i)\}$
where, $(L_c/m - i)$ is maximum number of chunks that peer D wants to download at that

specific time slot.

Next, we define $F^{c2v}(H_v, i', k)$, as the probability that a randomly selected cooperating peer which has i useful chunks in its buffer, in a specific time slot, looking for $i' = (L_c/m - i)$, chunks in that time slot, sends k requests to its neighbors. $F^{c2v}(H_v, i', k)$ can be recursively calculated as follows:

$$F^{c2v}(h_v, i', k) = U_i^{c2v} F^{c2v}(h_v - 1, i' - 1, k - 1) + (1 - U_i^{c2v}) F^{c2v}(h_v - 1, i', k). \quad (22)$$

where, $i' = (L_c/m - i)$, $0 \leq h_v \leq H_v$, $0 \leq k \leq \min\{i', h_v\}$ and $0 \leq i \leq L_c$,
The first component of the right hand side in (22) assumes the peer sends one of the requests to one of the H_v viewing neighbors. Hence, it is looking for one less chunk (i.e., $i' - 1$) from the remaining neighbors (i.e., $h_v - 1$) and sends one less request (i.e., $k - 1$).

The recurrence relation (22) has the following initial conditions:

$$F^{c2v}(h_v, i', k) = \begin{cases} 0 & \text{if } (h_v = 0 \text{ or } i' = 0) \text{ and } k > 0 \\ 1 & \text{if } (h_v = 0 \text{ or } i' = 0) \text{ and } k = 0. \end{cases}$$

Next, we calculate the expected-value of the number of requests sent by a cooperating peer having i useful chunks as $\overline{L}_i^{c2v}$ and define $\overline{k}^{c2v}$ as the average number of requests that any arbitrary cooperating peer sends to its viewing neighbors at any given time slot.

$$\overline{L}_i^{c2v} = \sum_{k=0}^{\min\{i', H_v\}} k F^{c2v}(H_v, i', k). \quad (23)$$

$$\overline{k}^{c2v} = \sum_{i=0}^{L_c/m} (P_i^c \overline{L}_i^{c2v}). \quad (24)$$

where, $i' = L_s - i$ We define $\overline{X}^{c2v}$ as the average number of requests that a randomly selected viewing peer (e.g., peer A) receives from its neighbors in addition to the request received from cooperating peer D:

$$\overline{X}^{c2v} = \frac{(H_c - 1)\overline{k}^{c2v}}{H_v} + \overline{k}^{v2v}. \quad (25)$$

The first term in (25) is related to the cooperating peers and it shows that H_c cooperating peers send $(H_c - 1)\overline{k}^{c2v}$ requests to their viewing neighbors and on an average each of the H_v viewing neighbors receive $(H_c - 1)\overline{k}^{c2v}/H_v$ requests from them.
The second term in (25) is related to the viewing peers and it shows that H_v cooperating

peers send $H_v\overline{k}^{v2v}$ requests to their viewing neighbors and on an average each of the H_v viewing neighbors receive $\overline{k}^{c2v}$ requests from them.
We define Q^{c2v} as the probability that a randomly selected peer (e.g., peer A) arbitrarily fulfills u_v requests (i.e., including the one from peer D) among all the requests it received

$$Q^{c2v} = \frac{u_v}{1 + \overline{X}^{c2v}}. \tag{26}$$

Upload capacity, u_v, is assumed to be 1. Hence, in the absence of requests from other neighbors, the only request from peer D is guaranteed to be fulfilled. Now, we compute the probability of forward transition in the discrete-time stochastic model shown in Figure 7 as follows:
We define $r^{c2v}_{i,n}$, as the probability that a cooperating peer, which has i useful chunks at a given time slot, downloads n chunks in the same time slot. Hence:

$$r^{c2v}_{i,n} = \sum_{k=n}^{\min\{\frac{L_c}{m}-i, H_v\}} F^{c2v}(H_v, \frac{L_c}{m} - i, k)\left(\binom{k}{n}(Q^{c2v})^n(1-Q^{c2v})^{k-n}\right). \tag{27}$$

Next, we define, M^{c2v}_i, probability that a randomly picked cooperating peer in the network having i useful chunks uniformly distributed in its cooperating buffer L_c at a specific time slot, can play a chunk in the same time slot. Since a cooperating peer does not playback a chunk, it simply advances its play (track) pointer at the rate of one chunk/time slot. This is how the cooperating buffer of a cooperating peer is updated the same way the viewing buffer of a viewing peer is updated with respect to time. In fact this is associated with the probability of transition from state i to state $i-1$ as shown in Figure 7.
Since we do not set priority to any of the chunks, the probability that we have the first chunk (for continuous playing or tracking) is:

$$M^{c2v}_i = \frac{i}{L_c/m} \qquad i = 0, 1, ..., L_c/m. \tag{28}$$

If the system is in state i, the probability the system jumps to state $i+k$ in the next time slot is given by:

$$B^{c2v}_{i,i+k} = (1 - M^{c2v}_i)r^{c2v}_{i,k} + M^{c2v}_i r^{c2v}_{i,k+1}. \tag{29}$$

If the system is in state i, the probability the system jumps to state $i-1$ in the next time slot is given by:

$$B^{c2v}_{i,i-1} = M^{c2v}_i r^{c2v}_{i,0}. \tag{30}$$

where, $k = 0, 1, 2, ... \min\{H_v, L_c/m\}$ In equation 29, the first term on the right hand side corresponds to the situation where the first chunk is missing (i.e., peer cannot playback in that time slot) and k chunks are downloaded. However, the second term represents a situation where the peer can play back in the current time slot and downloads $(k+1)$ chunks. In equation 30, the peer plays back in the current time slot but downloads no useful chunk. When the system is in steady state, the probability distribution of any state in the model (shown in Figure 7) does not change with time and we can write the global balance equation:

$$(B^{c2v}_{i+1,i}P^{c}_{i+1}) + \left(\sum_{k=1}^{\min\{H_v,i\}} (B^{c2v}_{i-k,i}P^{c}_{i-k})\right) - \left(\sum_{k=1}^{L_s-i}(B_{i,i+k}P^{c}_{i})\right) - (B^{c2v}_{i,i-1}P^{c}_{i}) = 0. \quad (31)$$

It is difficult to find a closed form of solution for the relation in (31) in order to get the peer distribution $\{P^c_i\}$. This is because, in the stochastic model, we have $L_s + 1$ unknown states and L_s+1 equations for the solution. Implementing such an analytical solution scheme for so many unknowns would cause memory leak in the system. However, it is possible to solve (31) numerically and results very close to the close form of solution. We compared our numerical solution with the simulation result from our own simulator.

Next, we define P^c_{cont}, as the probability that a randomly picked cooperating peer in the network that is keeping track of a certain live streaming content in a specific time slot, would be able to track the desired chunk at the same time slot. It can be expressed as follows:

$$P^c_{\text{cont}} = \sum_{i=0}^{L_c/m} P^c_i M^{c2v}_i. \quad (32)$$

Let d^{c2v}_i be the average download rate of a cooperating peer having i useful chunks in its buffer L_c/m in a specific time slot:

$$d^{c2v}_i = \sum_{k=0}^{\min\{H_v,L_c/m-i\}} \left(k r^{c2v}_{i,k}\right) \qquad i = 0, 1, ..., L_c/m. \quad (33)$$

The average download rate of a cooperating peer can be expressed as follows:

$$d^{c2v} = \sum_{i=0}^{L_c/m} \left(d^{c2v}_i P^c_i\right). \quad (34)$$

Figure 7 shows different states of A discrete-time Stochastic model for an arbitrarily selected cooperating peer in the HnH scheme based multi-channel live streaming system.

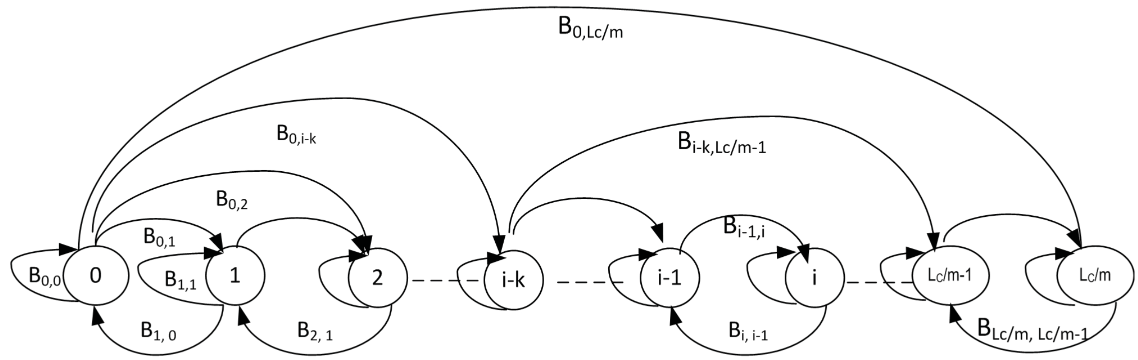


Figure 7: A discrete-time stochastic model for cooperating peers in the HnH scheme having effective buffer size L_c/m, where m is the number of sub-streams.

2.5.5 Probabilities of Requesting and Downloading Chunks for a Viewing Peer in HnH

We now assume that a peer will have both viewing and cooperating neighbors at the same time. Now, we define $F^{HnH}(H_v, H_c, i', k_v, k_c)$, as the probability that a randomly selected peer (e.g., peer A) having i useful chunks in its buffer, in a specific time slot, looking for $i' = L_s - i$, chunks in that time slot, sends k_v requests to its H_v viewing neighbors and k_c requests to its H_c cooperative neighbors.

$F^{HnH}(H_v, H_c, i', k_v, k_c)$, can be calculated from the following two recursive expressions:

$$F^{HnH}(h_v, h_c, x, k_v, k_c) = U_i^{v2v} F^{HnH}(h_v - 1, h_c, x - 1, k_v - 1, k_c) + (1 - U_i^{v2v}) F^{HnH}(h_v - 1, h_c, x, k_v, k_c). \quad (35)$$

The first component of the right hand side in (35) assumes the peer receives one of the requested chunks from one of the h_v viewing neighbors. Hence, in the second part, it is looking for one less chunk (i.e., $i' - 1$) from the remaining viewing neighbors (i.e., $h_v - 1$) and sends one less request (i.e., $k_v - 1$). and

$$F^{HnH}(0, h_c, x, 0, k_c) = U_i^{v2c} F^{HnH}(0, h_c - 1, x, 0, k_c - 1) + (1 - U_i^{v2c}) F^{HnH}(0, h_c - 1, i'_c, 0, k_c). \quad (36)$$

where, $0 \leq x \leq i'$, $0 \leq h_v \leq H_v$, $0 \leq h_c \leq H_c$, $0 \leq (k_v + k_c) \leq \min\{i', (H_v + H_c)\}$, $0 \leq k_v \leq \min\{x, h_v\}$, $0 \leq k_c \leq \min\{(x - k_v), h_c\}$ and $0 \leq i \leq L_s$. The first and the second

portion of (36) can be explained similar to (35).
The recursive equation (35) has the following initial conditions:

$$F^{HnH}(h_v, h_c, x, k_v, k_c) = \begin{cases} 1 & \text{if } (h_v + h_c = 0 \text{ or } x = 0) \ \& \ k_v = k_c = 0 \\ 0 & \text{if } (h_v + h_c = 0 \text{ or } x = 0) \ \& \ k_v + k_c > 0 \end{cases}$$

Next, we calculate the expected-value of sending k_v requests to H_v viewing neighbors as $\overline{L}_i^{v2v}$ and define $\overline{k}^{v2v}$ as the average number of requests that any randomly selected viewing peer sends to its viewing neighbors at any given time slot.

$$\overline{L}_i^{v2v} = \sum_{k_v=0}^{k_v^{max}} \sum_{k_c=0}^{\min\{(i'-k_v), H_c\}} k_v F^{HnH}(H_v, H_c, i', k_v, k_c). \tag{37}$$

$$\overline{k}^{v2v} = \sum_{i=0}^{L_s} (P_i^v \overline{L}_i^{v2v}). \tag{38}$$

where, $k_v^{max} = \min\{i', H_v\}$, $i' = L_s - i$
Now, we calculate the expected-value of sending k_c requests to H_c cooperating neighbors as $\overline{L}_i^{v2c}$ and define $\overline{k}^{v2c}$ as the average number of requests that any randomly selected viewing peer sends to its cooperating neighbors at any given time slot.

$$\overline{L}_i^{v2c} = \sum_{k_v=0}^{k_v^{max}} \sum_{k_c=0}^{\min\{(i'-k_v), H_c\}} k_c F^{HnH}(H_v, H_c, i', k_v, k_c). \tag{39}$$

$$\overline{k}^{v2c} = \sum_{i=0}^{L_s} (P_i^v \overline{L}_i^{v2c}). \tag{40}$$

where, $k_v^{max} = \min\{i', H_v\}$, $i' = L_s - i$
We define the followings:
(i) $\overline{X}^{v2v}$ as the average number of requests that a randomly selected viewing peer (e.g., peer B) receives from its neighbors in addition to the request received from peer A
(ii) Q^{v2v} as the probability that a randomly selected viewing peer (e.g., peer B) arbitrarily fulfills u_v requests (i.e., including the one from peer A) among all the requests it received:

$$\overline{X}^{v2v} = \frac{H_c \overline{k}^{c2v}}{H_v} + \frac{(H_v - 1)\overline{k}^{v2v}}{H_v}. \tag{41}$$

The first term in (41) is related to the cooperating peers and it shows that H_c cooperating

peers send $H_c\overline{k}^{c2v}$ requests to their viewing neighbors and on an average each of the H_v viewing neighbors receive $H_c\overline{k}^{c2v}/H_v$ requests from them.
The second term in (41) is related to the viewing peers and it shows that $H_v - 1$ viewing peers send $(H_v - 1)\overline{k}^{v2v}$ requests to their viewing neighbors and on an average each of the H_v viewing neighbors receive $(H_v - 1)\overline{k}^{v2v}/H_v$ requests from them.

$$Q^{v2v} = \frac{u_v}{1 + \overline{X}^{v2v}}. \tag{42}$$

Similarly, we define the followings:
(i) $\overline{X}^{v2c}$ as the average number of requests that a randomly selected cooperating peer (e.g., peer D) receives from its neighbors in addition to the request received from peer A
(ii) Q^{v2c} as the probability that a randomly selected cooperating peer (e.g., peer D) arbitrarily fulfills u_c requests (i.e., including the one from peer A) among all the requests it received:

$$\overline{X}^{v2c} = \frac{(H_v - 1)\overline{k}^{v2c}}{H_c}. \tag{43}$$

$$Q^{v2c} = \frac{u_c}{1 + \overline{X}^{v2c}}. \tag{44}$$

Now, we compute the probability transition in the discrete-time stochastic model shown in Figure 2 as follows:
We define, $r_{i,n}^{HnH}$, probability that a viewing peer having i useful chunks at a given time slot, downloads n chunks in the same time slot. Hence, depending on the type of neighbors (i.e., viewing and cooperating), $r_{i,n}^{HnH}$ can be expressed as follows:

$$r_{i,n}^{HnH} = \sum_{n_v=0}^{n} \sum_{k_v=n_v}^{S^{v2v}} \sum_{k_c=(n-n_v)}^{S^{v2c}} F^{HnH}(H_v, H_c, i', k_v, k_c) \left(\binom{k_v}{n_v}(Q^{v2v})^{n_v}(1-Q^{v2v})^{k_v-n_v}\right) \left(\binom{k_c}{n-n_v}(Q^{v2c})^{n_c}(1-Q^{v2c})^{k_c-n_c}\right). \tag{45}$$

where, $S^{v2v} = \min\{H_v, (L_s - i)\}$, $S^{v2c} = \min\{H_c, ((L_s - i) - k_v)\}$, $n_c = n - n_v$, $i' = L_s - i$.
In this analysis, we assume that the desired chunks, which are scheduled to be downloaded in the current time slot, gets downloaded and stored in the buffer of that peer in the same time slot. We also assume that the streaming chunks are uniformly distributed across the buffer of a randomly selected peer in the network.
Next, we define, M_i^{HnH}, probability that a randomly picked peer in the network with the buffer L_s and i uniformly distributed useful chunks, in its buffer in a specific time slot, can play a chunk in the same time slot. Since we do not set priority to any of the chunks, the

probability that we have a chunk at the first place is:

$$M_i^{HnH} = \frac{i}{L_s} \qquad i = 0, 1, ..., L_s. \tag{46}$$

If the system is in state i, the probability the system jumps to state $i + k$ in the next time slot is given by:

$$B_{i,i+k}^{HnH} = (1 - M_i^{HnH})r_{i,k}^{HnH} + M_i^{c2v} r_{i,k+1}^{HnH}. \tag{47}$$

The first term on the right hand side corresponds to the situation where the first chunk is missing (i.e., peer cannot playback in that time slot) and k chunks are downloaded. However, the second term represents a situation where the peer can play back in the current time slot and downloads $(k + 1)$ chunks. If the system is in state i, the probability the system jumps to state $i - 1$ in the next time slot is given by:

$$B_{i,i-1}^{HnH} = M_i^{HnH} r_{i,0}^{HnH}. \tag{48}$$

where, $k = 0, 1, 2, ... \min\{H_v + H_c, L_s\}$
In equation 48, the peer plays back in the current time slot but downloads no useful chunk. When the system is in steady state, the probability distribution of any state in the model (shown in Figure 2) does not change with time and we can write the global balance equation:

$$(B_{i+1,i}^{HnH} P_{i+1}^v) + \left(\sum_{k=1}^{\min\{(H_v+H_c),i\}} (B_{i-k,i}^{HnH} P_{i-k}^v) \right) - \left(\sum_{k=1}^{L_s-i} (B_{i,i+k} P_i^v) \right) - (B_{i,i-1}^{HnH} P_i^v) = 0. \tag{49}$$

It is difficult to find a closed form of solution for equation 49 in order to get peer distribution $\{P_i^v\}$. This is because, in the stochastic model, we have $L_s + 1$ unknown states and $L_s +$ 1equation for the solution. However, it is possible to solve (49) numerically with less difficulty and results very close to the close form of solution. We compared our numerical solution with the simulation result from our own simulator. Next, we define $P_{\text{cont}}^{\text{HnH}}$, as the probability that a randomly picked peer in the network under HnH scheme that is watching or listening to a certain live streaming content in a specific time slot, would be able to play its desired chunk at this time slot. It can be expressed as follows:

$$P_{\text{cont}}^{\text{HnH}} = \sum_{i=0}^{L_s} P_i^v M_i^{HnH}. \tag{50}$$

Let d_i^{HnH} be the average download rate of a peer that has i useful chunks in its buffer L_s in a specific time slot:

$$d_i^{HnH} = \sum_{k=0}^{\min\{H, L_s - i\}} \left(k r_{i,k}^{HnH} \right) \qquad i = 0, 1, ..., L_s. \tag{51}$$

Then the average download rate of a peer can be expressed as follows:

$$d^{HnH} = \sum_{i=0}^{L_s} \left(d_i^{HnH} P_i^v \right). \tag{52}$$

2.5.6 Consumption of Effective Upload bandwidth

In this section we calculate the effective upload bandwidth consumed (or upload cost) of a peer before and after addition of cooperating peers. We recall Figure 1 where interactions among the peers in the HnH scheme is shown. As described in Section 2.3, a peer in the HnH scheme may receive a request to upload a chunk from three different types of interaction scenarios: i) Viewing-to-Viewing scenario (V2V) ii) Viewing-to-Cooperating scenario (V2C) iii) Cooperating-to-Viewing scenario (C2V). Figure 8 and 9 shows the upload and download components of a peer. In the subsequent calculations, we consider Figure 8 and the scenarios, (i.e., V2V, V2C, C2V) are considered in the context of uploading direction but not downloading.

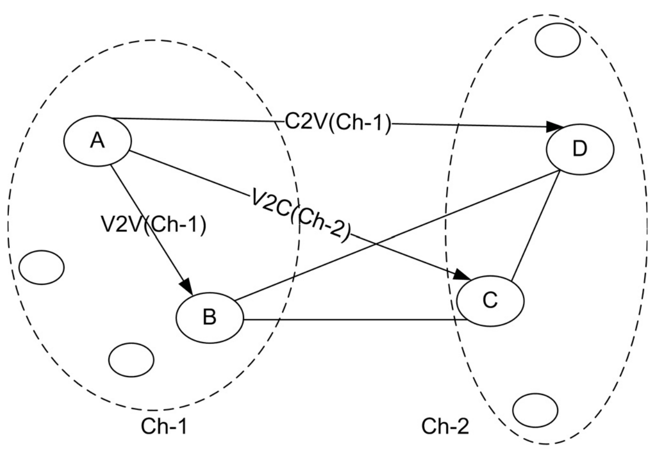


Figure 8: Different components of the upload BW of a peer in the HnH scheme

Viewing-to-Viewing scenario(V2V):
In this scenario, a viewing peer A uploads to its viewing neighbor (e.g., peer B) in response to the request for the chunks of the viewing channel of A. We recall from Section 2.5.5, the probability of sending and downloading chunks from neighbors in (35) and (36).

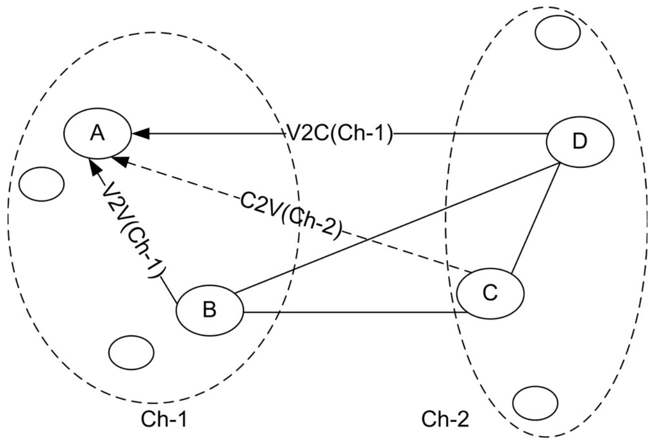


Figure 9: Different components of the download BW of a peer in the HnH scheme

Next, we recall the expected-value of sending k_v requests to H_v viewing neighbors in HnH scheme as $\overline{L}_i^{v2v}$ and the average number of requests that any randomly selected viewing peer sends to its viewing neighbors at any given time slot, as $\overline{k}^{v2v}$.

$$\overline{L}_i^{v2v} = \sum_{k_v=0}^{k_v^{max}} \sum_{k_c=0}^{\min\{(i'-k_v),H_c\}} k_v F^{HnH}(H_v, H_c, i', k_v, k_c). \tag{53}$$

$$\overline{k}^{v2v} = \sum_{i=0}^{L_s} (P_i^v \overline{L}_i^{v2v}). \tag{54}$$

where, $k_v^{max} = \min\{i', H_v\}$, $i' = L_s - i$
Now, we define x^{v2v} as the probability that a viewing peer (e.g., peer A) receives one request from any randomly selected neighbor (e.g., peer B) at any given time slot.
P(peer A receives one request from its viewing neighbor B) $= \overline{k}^{v2v} \ / \ H_v = x^{v2v}$
Next, we define $B^{v2v}(H_v, x^{v2v})$, probability that a peer (e.g., peer A) receives n_{v2v} request out of H_v neighbors at any given time slot is a binomial distribution with H_v and x^{v2v}, as follows:

$$P\left\{N^{v2v} = n_{v2v}\right\} = \binom{H_v}{n_{v2v}} (x^{v2v})^{n_{v2v}} (1 - x^{v2v})^{H_v - n_{v2v}} = B^{v2v}(H_v, x^{v2v}). \tag{55}$$

where, $0 \leq n_{v2v} \leq H_v$.
Now, we calculate these probabilities for V2C and C2V cases, in the similar manner.

Cooperating-to-Viewing scenario (C2V):

In this scenario, a viewing peer A uploads chunks of the viewing channel to its cooperating neighbor (e.g., peer D) in response to its request. We recall from Section 2.5.4, the probability of sending and downloading chunks by a cooperating neighbor in (22). We also recall the expected-value of the number of requests sent by a cooperating peer D to its cooperated neighbor A, as $\overline{L}_i^{c2v}$ and the average number of requests that any arbitrary cooperating peer sends to its viewing neighbors at any given time slot as $\overline{k}^{c2v}$.

$$\overline{L}_i^{c2v} = \sum_{k=0}^{\min\{i',H_v\}} kF^{c2v}(H_v, i', k). \tag{56}$$

$$\overline{k}^{c2v} = \sum_{i=0}^{L_c/m} (P_i^c \overline{L}_i^{c2v}). \tag{57}$$

where, $i' = L_c/m - i$

Now, we define x^{c2v} as the probability that a viewing peer (e.g., peer A) receives one request from any randomly selected cooperating neighbor (e.g., peer D) at any given time slot.

P(peer A receives one request from its cooperating neighbor D) $= \overline{k}^{c2v} \ / \ H_v = x^{c2v}$

Next, we define $B^{c2v}(H_c, x^{c2v})$, probability that a peer (e.g., peer A) receives n_{c2v} request out of H_c neighbors at any given time slot is a binomial distribution with H_c and x^{c2v}, as follows:

$$P\left\{N^{c2v} = n_{c2v}\right\} = \binom{H_c}{n_{c2v}} (x^{c2v})^{n_{c2v}} (1 - x^{c2v})^{H_c - n_{c2v}} = B^{c2v}(H_c, x^{c2v}). \tag{58}$$

where, $0 \leq n_{c2v} \leq H_c$.

Viewing-to-Cooperating scenario (V2C):

In this scenario, the role of a peer is just opposite to the role it played in V2V and V2C scenarios. Now, a cooperating peer of ch-1 (e.g., peer C) acts as a viewing peer for ch-2 and a viewing peer of ch-1 (e.g., peer A) acts as the cooperating neighbor for ch-2. Peer C (which now acts as a viewing peer for ch-2) sends request to one of its H_c cooperating neighbors, A (which now acts as a cooperating peer for C). We recall from Section 2.5.5, the probability of sending and downloading chunks from neighbors in (35) and (36).

Next, we recall the expected-value of sending k_c requests to H_c cooperating neighbors as $\overline{L}_i^{v2c}$ and the average number of requests that any randomly selected viewing peer sends to its cooperating neighbors at any given time slot, as $\overline{k}^{v2c}$.

$$\overline{L}_i^{v2c} = \sum_{k_v=0}^{k_v^{max}} \sum_{k_c=0}^{\min\{(i'-k_v),H_v\}} k_c F^{HnH}(H_v, H_c, i', k_v, k_c). \tag{59}$$

$$\overline{k}^{v2c} = \sum_{i=0}^{L_s} (P_i^v \overline{L}_i^{v2c}). \tag{60}$$

where, $k_c^{max} = \min\{i', H_c\}$, $i' = L_s - i$

Now, we define x^{v2c} as the probability that a cooperating neighbor (e.g., peer A) receives one request from any randomly selected cooperated peer (e.g., C) at any given time slot.

P(peer A receives one request from its cooperated neighbor C) $= \overline{k}^{v2c}$ / $H_c = x^{v2c}$

Next, we define $B^{v2c}(H_c, x^{v2c})$, probability that a peer (e.g., peer A) receives n_{c2v} request out of H_c neighbors at any given time slot is a binomial distribution with H_c and x^{c2v}, as follows:

$$P\left\{N^{v2c} = n_{v2c}\right\} = \binom{H_c}{n_{v2c}} (x^{v2c})^{n_{v2c}} (1 - x^{v2c})^{H_c - n_{v2c}} = B^{v2c}(H_v, x^{v2c}). \tag{61}$$

where, $0 \leq n_{v2v} \leq H_v$.

Total effective upload capacity without adding cooperating peers:

Before adding any cooperating peer to the HnH scheme, the effective upload capacity was contributed only towards the viewing neighbors. We define N^{v2v} as the total number of requests that peer A receives from its viewing neighbors.

$$N^{v2v} = P\{N = n_{v2v}\} = \binom{H_v}{n_{v2v}} (x^{v2v})^{n_{v2v}} (1 - x^{v2v})^{H_v - n_{v2v}}. \tag{62}$$

Total effective upload capacity with cooperating peers in the HnH scheme:

In the HnH scheme, the effective upload capacity is contributed to three different scenarios. We define N^{HnH} as the total number of requests that peer A receives, which includes: *(i)* requests from its viewing neighbors (same channel), *(ii)* requests from its cooperating neighbors(same channel) *(iii)* requests from the cooperated neighbors (i.e., viewing peers of a different channel for whom peer A is currently acting as a cooperating peer.) Hence, total number of requests can be written as: $N^{HnH} = N^{v2v} + N^{c2v} + N^{v2c}$. We can calculate the probability that a peer (e.g., peer A) receives N^{HnH} requests, using convolution. However, we do not show the detailed calculation here. Now, we calculate the effective upload capacity

u_{eff} from the maximum allowable upload capacity $u_{\max}$ as follows:

$$u_{\text{eff}} = \left[\sum_{k=1}^{u_{\max}-1} kP\{N = k\} \right] + u_{\max} P\{N > u_{\max}\}. \qquad (63)$$

2.6 Simulation and Numerical Results

In order to validate the proposed HnH scheme, we first get some numerical results (i.e., probability of continuity) without helping peers. Next, we introduce helping peers and then compare the probability of continuity with and without helping peers. In addition, in order to validate the proposed stochastic model, we compare the probability of continuity, P_{cont}, derived from the stochastic model with the probability of continuity obtained from our simulator. Moreover, we show improvement in downloading the chunks when helping peers are added. Finally, we compare the effective upload Bandwidth consumption, u_{eff} before and after the cooperating peers are added in the proposed HnH scheme.

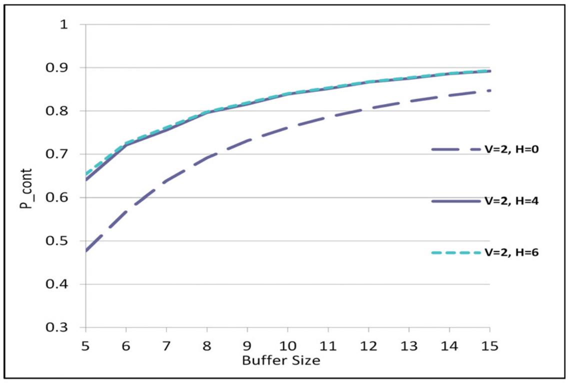
2.6.1 Experimental Parameters

We assume that time is slotted and first set the normalized upload capacity, $u = 1$ chunk/time slot (later we will show cases where $u > 1$). In the experimental setup, a streaming file at the source is divided into $N = 500$ chunks, and $m = 4$ sub-streams. Each viewing peer in the HnH overlay is connected to a maximum of $H = 5$ viewing neighbors. Playback buffer size, L_s, of a viewing peer and the maximum playback delay, T, is considered in the range of: $2 \leq L_s \leq 15$ and $2 \leq T \leq 15$.

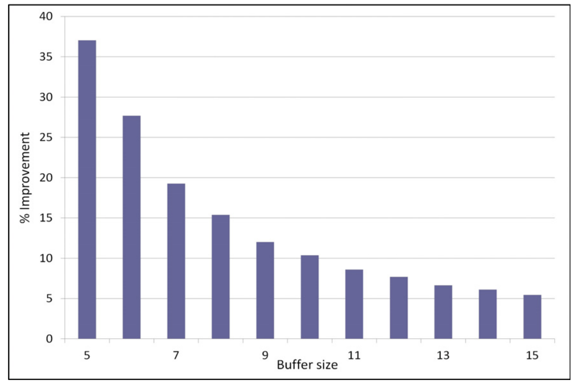
2.6.2 Effect of Adding Helping Peers

We now analyze the probability of continuity against different sizes of playback buffers before and after helping peers are added. Results are summarized in Figure 10(a). The dotted line at the bottom shows the probability of continuity when a peer from a DCSV channel has only $V = 2$ viewing neighbors. The effect of adding different number of helping peers on the probability of continuity are represented by the dotted lines above.

Figure 10(b), shows the percentage improvement in probability of continuity when $H = 4$ helping peers from a different DCSV channel are added to the existing viewing peers in the current DCSV channel.



(a) P_{cont} before/after adding helping peers



(b) % Improvement in P_{cont}

Figure 10: Impact on P_{cont} after adding helping peers

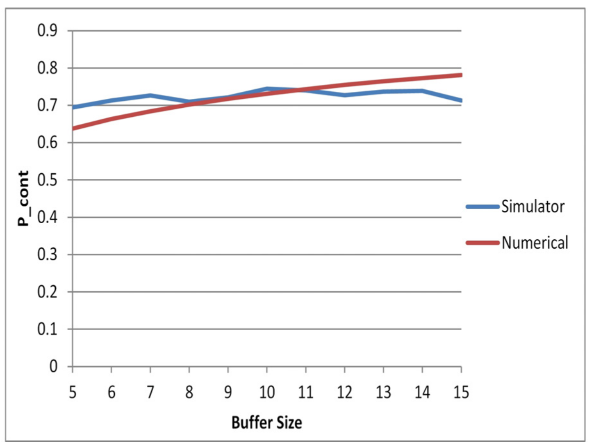


Figure 11: Comparing simulation and stochastic model results

Now, Figure 11 shows the comparison between the probability of continuity, P_{cont}, from the simulator and P_{cont} derived from the stochastic model. From this, we find that the stochastic model assumes realistic assumptions.

Figure 12, shows average download rate against the number of available (i.e., useful) chunks in the buffer before an after the cooperating peers are added. We find that the average download rate of peers have increased significantly.

2.6.3 Effect of Maximum Delay from the Source

Here we analyze the effect of maximum delay from the source on probability of continuity, P_{cont}. Figure 13, shows probability of continuity P_{cont} against maximum delay T for three different values of buffer size $L_s = \{5, 10, 15\}$. The line at the bottom represents P_{cont} due

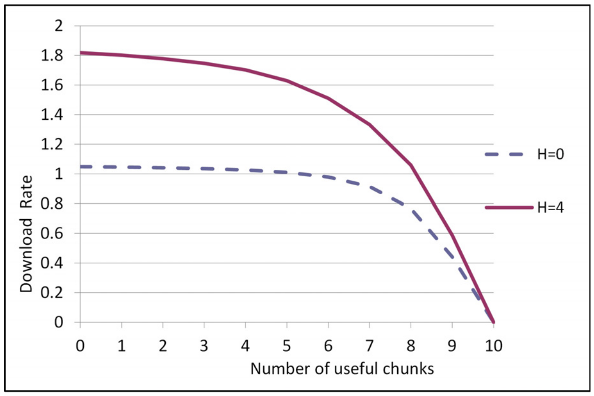


Figure 12: Improvement in downloading of chunks when helping peers are added

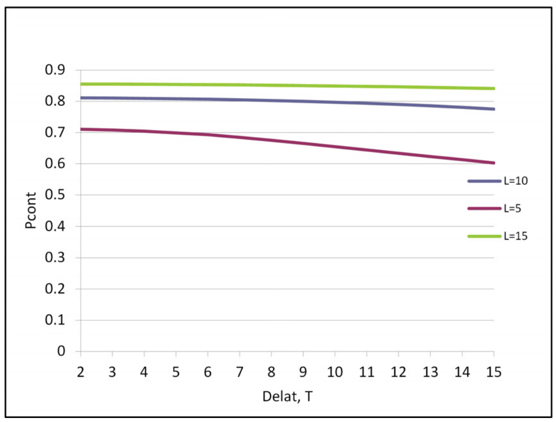


Figure 13: Probability of continuity (P_{cont}) decreases when maximum delay is increased

to $L_s = 5$ and the one at the top due to $L_s = 15$.

2.6.4 Effect of Adding more Viewers from the Same Channel

In this section we analyze the effect on probability of continuity P_{cont} if we could have added more viewing peers from the same channel. However, in practice, we are unable to add more viewers to a DCSV channel. Figure 14, shows the effect of different number of viewers on the probability of continuity P_{cont} against buffer size L_s and maximum upload capacity $u_{\text{max}} = 2.0$. The dotted line at the bottom represents number of viewer $v = 5$ and the one at the top $L_s = 15$.

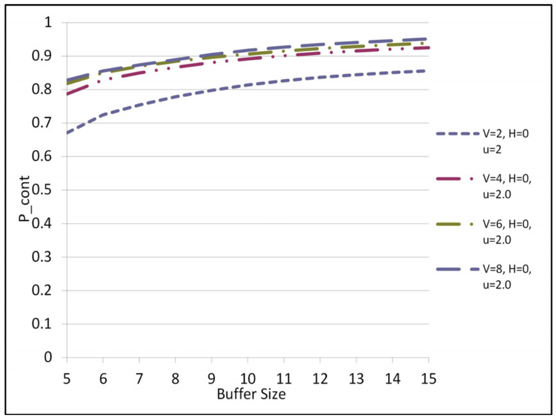


Figure 14: Effect of adding viewers from the same channel

2.6.5 Effect of Maximum Upload Capacity and Effective Upload Capacity of a Peer

We analyze here the effect of increasing the maximum upload capacity on probability of continuity, P_{cont}.

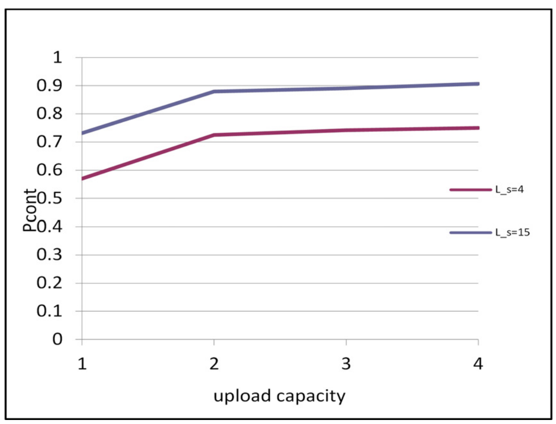
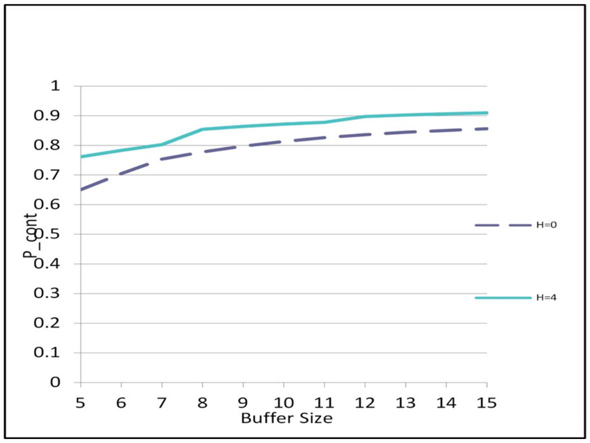


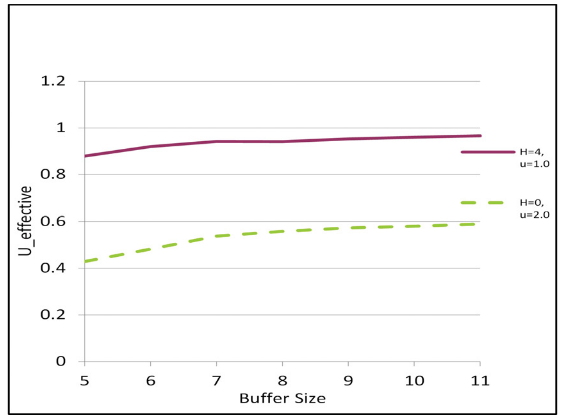
Figure 15: Probability of continuity vs. peers' upload capacity

Figure 15 depicts the effect of increasing upload capacity u on the probability of continuity P_{cont} for different buffer size L_s with a number of viewers equal to 5.

Figure 16(a) compares the improvement of Probability of continuity P_{cont} when helping peers are added to small number of viewers with small maximum upload capacity, $u_{\max} = 1.0$ with the case where upload capacity is increased to $u_{\max} = 2.0$ without adding helping peers. Figure 16(b) compares the corresponding consumption of effective upload BW, u_{eff}. We make the following observations from the experimental results:
(i) The probability of continuity is significantly improved when helping peers from another DCSV channel are added to the viewers of the current DCSV channel. Figure 10(b) suggests



(a) Probability of continuity P_{cont}



(b) Consumption of effective upload bandwidth, u_{eff}

Figure 16: Comparison of probability of continuity P_{cont} and consumption of effective upload bandwidth, u_{eff} before adding helping peers (upload rate $\rightsquigarrow u$=2.0) and after adding helping peers (upload rate $\rightsquigarrow u$=2.0)

that adding helping peers obviously improves the performance in the current scenario.
(ii) We found that performance of the viewing peers significantly depends on the size of the playback buffer.
(iii) We found that the performance of the viewers does not depend much on the maximum delay T, from the source. As shown in Figure 13, the performance may even decrease for smaller buffer size
(iv) From Figure 14, we find that too high upload capacity (i.e., after $u > 2.0$) does not have significant effect on the playback continuity. We further find that even with doubled upload capacity, playback continuity is not satisfactory when the number of viewers is below 8.
(v) Figure 16(b) shows effective upload capacity used (or upload cost) with and without helping peers. The dotted line in this figure represents the amount of effective upload capacity used when the maximum allowed upload capacity is increased to Umax=2.0 (in order to increase the playback performance). The solid line represents the case where maximum allowed upload capacity is kept 1.0, however, a number of 4 helping peers are added to improve the performance.

From the above results, we find that even when the maximum allowed upload capacity is increased to higher value (i.e., Umax = 2.0), it cannot be fully utilized to improve the playback performance of a peer due to having small number of neighbors. On the other hand, playback performance can be improved by adding helping peers (by making efficient use of the upload capacity) even when the maximum allowed upload capacity remains low (e.g., Umax = 1.0). Using our stochastic model, we showed how probability of continuity (i.e., probability of continuous playback) using the proposed cross channel resource sharing scheme in the multi-channel HnH based system can be significantly improved. Standard compressed video (e.g, MPEG4, 480p resolution) uses bit rate of 600 kbps. In most of the residential DSL/Cable modems, available upload capacity is in the range of 800 kbps to 1.0 mbps which, in our model, is equivalent to the range of u=1.5 chunks to u=2.0 chunks/time slot. Again, the download capacity in DSL/Cable modems is usually in the range of 5 mbps to 6 mbps which is equivalent to d=10 chunks/time slot. These justifies our assumption of not considering bandwidth bottleneck on the download capacity. It also shows that cooperating peers can exchange additional cooperating chunks within available upload capacity without affecting their own performance.

Parameter	minimum	Maximum
NoOf Nodes	10	50
NoOf Viewers	5	10
NoOf Helpers	5	10
Playback Buffer, L_s	2	15
NoOf sub-streams	2	5
Maximum Delay, T	2	15

Table 1: Key Parameters for HnH Scheme

2.7 Conclusions and Future Work

In this work, we showed how probability of continuity (i.e., probability of continuous playback) using the proposed cross channel resource sharing scheme in the multi-channel HnH system can be significantly improved. We have introduced cooperation among the peers from different DCSV channels where every peer is very likely to suffer from poor channel performance. We believe that this sufferings induce a sufficient motivation for all the peers to cooperate, even if it means downloading chunks from a channel, they are not interested in viewing. Especially, since this cooperation ultimately improves the performance of their own channel, this becomes a strong incentive for the peers across DCSV channels to help each other. We developed a simple discrete-time stochastic model for DCSV channels in a HnH system which can provide guideline for efficient designing of DCSV channels. We analyzed the impacts of different parameters (e.g., buffer size, maximum delay, number of helping peers etc.) on the performance. Beside the proposed HnH cooperation scheme, the core contribution of this paper is the discrete-time stochastic model in order to evaluate the performance based on collaboration among peers of DCSV channels of a HnH system. To the best of our knowledge, this is the first work where the performance problem of DCSV channels is addressed by considering the content bottleneck issue instead of focusing on the bandwidth bottleneck issue. Our proposed HnH scheme relies on natural incentive for co-operation among the performance deprived peers of DCSV channels who are assumed to be naturally interested to help each other for better performance. Future work of this topic may include: *(i)* Devising an efficient incentive mechanism for a sustainable HnH system, *(ii)* Designing an efficient content retrieval mechanism for the DCSV channels in a HnH system.

Chapter 3

Probability Model for Live Streaming Channels having Free riders with and without Incentive Mechanism

3.1 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 [2], PPLive [3], Sopcast [4]. Thousands of users can simultaneously participate in these systems. Almost all Live P2P video systems accommodate multiple channels (e.g., PPLive [3] can host over hundreds of channels). It is expected that, in the near future, Live streaming systems with several hundreds of user-generated channels and dedicated channels will likely have thousands of Live channels in total. A common practice in the Live streaming systems is to organize the peers viewing the same Live channel into a swarm where they form a mesh-based structure and distribute/re-distribute the streaming pieces (commonly known as chunks) to each other. Swarm formations work well when the participant peers responds generously to the upload requests of its neighbors. On the contrary, peers may experience severe degradation of continuous playback quality if many of the peers act selfishly as Free riders. We propose that presence of a simple incentive mechanism will improve the continuous playback quality of the peers. First, we develop a discrete-time stochastic model for the peers of Live P2P video streaming. We compare network parameters like probability of continuous playback, average download rate etc. with certain amount of Free riders present in the system as well as without any Free rider. Next, we introduce a simple incentive mechanism and modify our stochastic model in order to accommodate the incentive mechanism. Then we compare the result of probability of continuous playback with and without having an incentive mechanism. Our work shows that presence of an incentive mechanism improves the over all system performance. This paper is organized as follows. In Section 3.2, we discuss the relevant recent work from the literature.

3.2 Related Work

Free riders have been studied for long from different perspectives in different contexts. Free rider is first noted to be discussed by E.Ader *et al.* [23] in P2P context, in which the authors pointed out that there are many Free riders in the network who try to take advantage of the system. Several works have been published on Free riders problem in P2P File sharing systems where as there are few works done in the context of Live streaming system. Next, we present some of the works found in literature. Free riders have been studied for File sharing systems, e.g., Bittorent network by Qiu *et al.* [24], and for single channel Live streaming systems as well, e.g., in Liu *et al.* [25]. Qiu *et al.* have shown that even if the tit-for-tat mechanism is very good to protect a network from Free riders in a File sharing system, the Bittorent protocol still has one option called 'unchoking' which is exploited by the Free riders for their benefit. Qiu *et al.* have suggested a modification in the Bittorent protocol, more specifically in the unchocking technique and have shown this modified unchoking technique can prevent Free riders. Zhengye *et al.* [25] have suggested multiple layered coding as an incentive mechanism to avoid the tendency of Free riding. In this mechanism, the viewing quality of the streaming video received by a user is be determined by the contribution of upload bandwidth of that user. Koo *et al.* [27] have suggested an incentive mechanism for a P2P Content distribution system, where the jobs may be divided into smaller units, and users have incentives to truthfully revealing willingness-to-pay for services. In this mechanism, the capacity that every user makes available to others in turn determines the amount of resources that user receives. However, this mechanism is designed for the P2P File sharing schemes. Habib *et al.* [28] have developed a score-based incentive mechanism for single channel media streaming. A peer which gets higher score is rewarded with more choice of neighbor selection and flexibilities. However, it does not address the issue of fulfilling their incentive mechanism honestly. Kumar *et al.* [29] have designed a pricing and allocation mechanism for a special case of P2P network that allows the users within a firm to effectively share their resources and avoid Free riding. Here, the optimal price of a task is determined based on the delay at each peer. Wu *et al.* [30] have developed an auction and bid based incentive mechanism for P2P File sharing systems in the context of social network using game theory. They also have suggested an extension of their model for real time Live streaming systems. Altman *et al.* [31] have developed a stochastic model to study behavior of the P2P networks those distribute non-authorized music, books, or articles. They even consider the presence of Free riders in that network. Then this model is used to predict the efficiency of the counter-measures taken by the content providers against those P2P networks. Zhao *et al.* [32] have developed an analytical model to study the behavior of the Video on Demand P2P network having Free riders in the system. Jin *et al.* [13] considered a game theoric approach to show the performance improvement when the peers in presence of the Free riders are motivated to cooperate more. However, they have only considered single channel scenario in their model. Shahriar *et al.* [21] have developed a generic stochastic model for peers cooperating with each other where these peers are from two different channels having small number of viewers. In [22] the authors have presented another stochastic model which depicts interactions among the peers from two different channels with greater details and more realistic assumptions. In [33] the authors have presented a new stochastic model which considers the presence of Free riders in live streaming systems. In this work, the authors show the impact of the Free riders on the playback performance of the peers of the system. However, their work does not discuss about the impact of incentive mechanism on the Free riders. Yeung *et al.* [34] have designed a tit-for-tat based incentive mechanism for single channel P2P media streaming and modeled this as two repeated game. In one of the repeated game, they have shown interactions between the streaming server and the immediate peers. The other one deals with the interaction between a peer and its neighbor. Although they have formulated an optimization model for the first case, they provided only the simulation results. However, none of the works mentioned above have analyzed the effect of Free riders with a stochastic model for Live streaming systems.

3.3 Description of the Stochastic Model where Free riders are Unidentified

In this section we developed a discrete-time stochastic model in order to investigate the effect of Free riders on the P2P live streaming systems. We describe this model below. In this scheme, a peer may have two types of neighbors: *(i)* Contributors: peers who behave generously and contribute chunks when requested and
(ii) Free Riders: peers who behave selfishly and do not contribute chunks when requested.

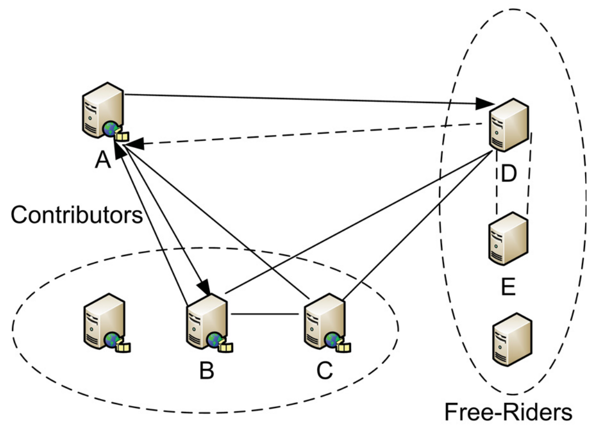


Figure 17: A Peer having contributors and Free Riders as neighbors

3.3.1 Outline of the Stochastic Model

Now, we discuss the Stochastic Model in details. We assume the peers to be homogeneous, i.e., to have identical parameters. The following notations are used in the stochastic model:
L_s : length of the viewing/playback buffer;
t_s : playback pointer (i.e., the sequence number of the chunk) at the source;
$(t_s - T_0)$: the earliest playback pointer at any peer with delay T_0 from the source;
$(t_s - T_1)$: the latest playback pointer at any peer with delay T_1 from the source;
$T = (T_1 - T_0 + 1)$: the maximum amount of delay between the playback pointer of the source and that of a peer;
u : the upload bandwidth of a peer;
H_c : total number of contributing neighbors of a peer;
H_f : total number of Free riding neighbors of a peer;
$H = (H_c + H_f)$: total number of neighbors of a peer;
P_f : probability that a peer is a Free rider;
$(1 - P_f)$: probability that a peer is a contributor (i.e., not a Free rider);
q_f : probability that a Free rider acts generously as a contributor;

The model relies on the following assumptions:
- Each peer maintains a fixed number of neighbors, H.
- The download capacity of any peer is unlimited but the upload capacity, u chunk/time slot, is limited and same for all peers.
- Each peer maintains a limited and fixed sized playback buffer, L_s, in its memory.
- A peer is said to have i useful chunks (for itself) in its playback buffer L_s, if none of these

i chunks have been played yet by this particular peer. These useful chunks cannot be overwritten by newer chunks before they are played.
- The playback rate of a peer is 1 chunk/time slot and its playback pointer denotes the sequence number of the chunk that the peer is playing at the current time slot. Due to limited size of the buffer, these $(L_s - i)$ old (i.e., already played) chunks can be overwritten by newer chunks.
- If a peer contains j useful chunks in its buffer L_s then $(L_s - j) \geq 0$ old chunks may also be available in the buffer at the same time.
- If a peer with i useful chunks is looking for $(L_s - i)$ chunks, all the missing $(L_s - i)$ chunks have equal priority to be fetched. More specifically, no priority is given to chunks close to the playback pointer. Studying the effect of such a priority is part of one of our future works.
- Upload bandwidth is set to u chunks/time slot, where $u \geq 1$ is a constant. However, download bandwidth is assumed to be unlimited since in most P2P systems, normally, download bandwidth is not a bottleneck issue.
The stochastic model (as illustrated in Figure 18) is described by $(L_s + 1)$ states (of a peer), where each state corresponds to its number of useful chunks in the playback buffer. Associated with each state i (except the initial one), there is exactly one reverse transition (M_i) to the its previous state (i.e., $i - 1$) which corresponds to the playback of one chunk/time slot. From each state i, there originates $(L_s - i)$ forward transitions, $B_{i,k}$, where $0 \leq k \leq (L_s - i)$. In the subsequent sections, we discuss these transition probabilities in details.
$(L_s - i) \geq k \geq 0$

3.3.2 Objective and Constraints:

The objective of the stochastic model is to investigate the fundamental characteristics, limitations and performance problems of a P2P Live streaming system having Free riders, depending on the values of the network parameters (e.g., buffer size, maximum delay, number of Free riders). After computing the transition probabilities, we will be able to estimate the probability distribution of the number of useful chunks. From these probability distributions, we calculate the probability of continuous playback and gain more information on how the continuous playback performance of such a P2P Live streaming system is affected by the amount of Free riders in addition to the network parameters (e.g., buffer size, maximum delay, number of Free riders). We have the following constraints:
- A streaming chunk, k, will be stored by peer A if it satisfies: $t_A \leq k \leq (t_A + L_s - 1)$ where, t_A is the playback pointer of A, otherwise, it will be discarded.
- A peer having i useful chunks in its buffer L_s sends a maximum of $\min\{H, (L_s - i)\}$ requests to its H neighbors in the current time slot.

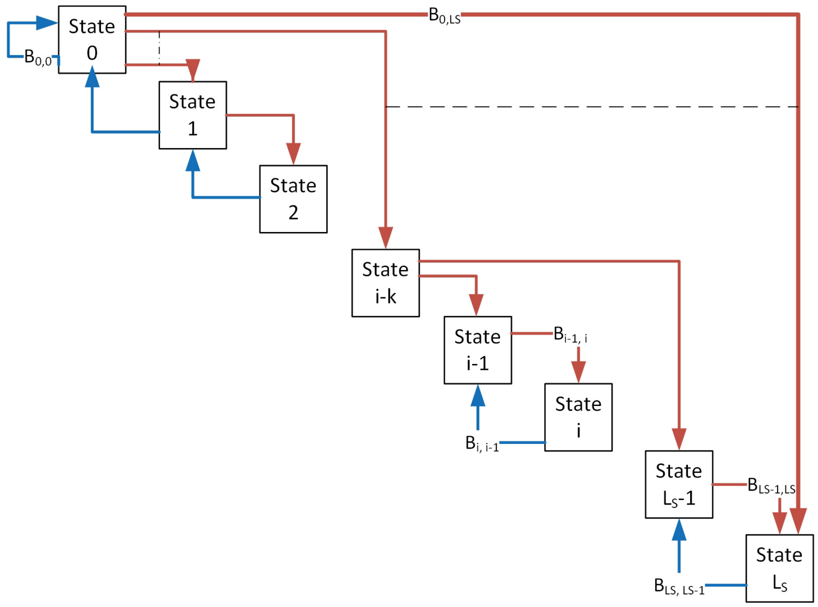


Figure 18: A discrete-time stochastic model for peers (i.e., both contributing and Free riding types) having buffer size L_s.

- The playback pointer t_A of peer A lies within the following limit: $(t_s - T_1) \leq t_A \leq (t_s - T_0)$.

3.4 Analysis of the Stochastic Model

Now, we analyze the stochastic model and calculate the probabilities for the scenarios mentioned in Section 4.4. In a Live streaming channel, the request of a peer to its neighbor in order to get a chunk, depends on the relative positions of their playback pointers in the corresponding playback buffers. We have the following two categories of positions:
(1) No overlapping of chunks between the corresponding buffers of peer A and its neighbor B and
(2) Some overlapping of chunks between the corresponding buffers of peer A and and its neighbor B. Category 1 can be further sub-divided into two sub-categories:
(1.1) Peer A is playing ahead of its neighbor B and there is no overlapping of chunks in their buffers (i.e., peer A has already played all the chunks of its neighbor B) and (1.2) Neighbor B is playing much ahead of peer A such that even the old chunks of B are too new for peer A to request (i.e., there is no overlapping). Category 2 can be further sub-divided into three sub-categories:
(2.1) Peer A is playing ahead of its neighbor B, having some overlapping chunks between A and B, (2.2) Neighbor B is playing ahead of peer A and there are some overlapping chunks between A and B and (2.3) Neighbor B is playing much ahead of peer A however, there is

some overlapping of old chunks of B with the new chunks of A. These five general categories are illustrated in Figure 19 and Figure 20. We next derive the probabilities of the stochastic model which are common to both C2N and F2N scenarios.

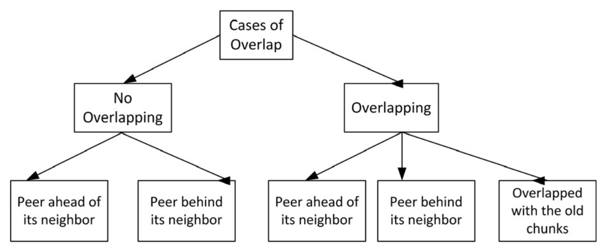


Figure 19: Cases of relative positions of the playback pointers in the buffers of peer A and its neighbor B.

3.5 Analysis of the Stochastic Model without any Incentive

In this model, we assume that currently there is no way to know which peer is a contributing peer and which peer is a Free riding peer. As illustrated in Figure 17, we identify the following interaction scenarios for a peer in this model:

- **Contributing-to-Neighbors (C2N) Scenario:** In this scenario, a contributing peer responds to the request of its neighbors (including both contributors and Free riders) by uploading chunks to them.
- **Free riding-to-Neighbors (F2N) Scenario:** In this scenario, a Free riding peer responds to the request of its neighbors (including both contributors and Free riders) by uploading (or not uploading) chunks to them.

Consider an arbitrary peer (e.g., peer A) and one of its neighbor (e.g., peer B) such that they may be of any combination of the contributing and Free riding peers. Suppose, at the current time slot, peer A has i and its neighbor B has j useful chunks in their respective buffers. Based on that, we will calculate, U_i, probability that peer A having i useful chunks

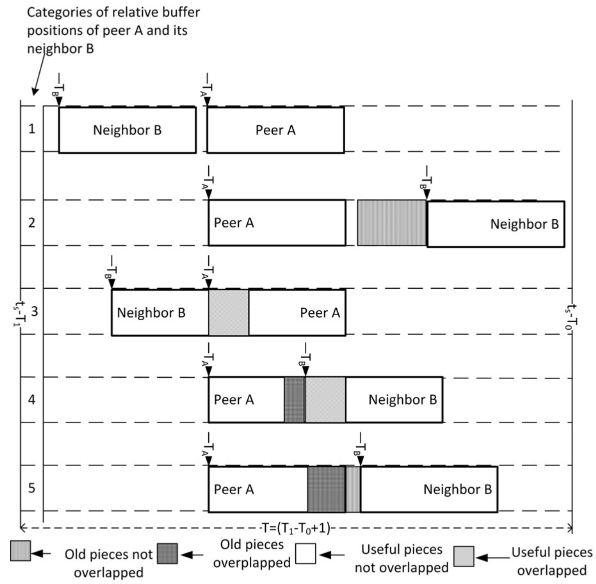


Figure 20: Relative positions of peer A and its neighbor B where t_A and t_B denote the playback pointer of the peer A and its neighbor B, respectively.

in its buffer in a specific time slot, will be interested to get a chunk from a random neighbor, B.

$$U_i = \sum_{j=0}^{L_s} (1 - f(i,j))\, P_j. \tag{64}$$

where P_j is the probability that a random contributing peer in the network has j useful chunks. In order to calculate, U_i, first we define $f(i,j)$, to be the probability that peer A with i useful chunks in its buffer is not interested in any of the j useful chunks of B. We calculate $f(i,j)$ for the following five mutually exclusive cases:

$\text{Case}_1 : t_s - T \leq t_B \leq t_A - L_s$

$\text{Case}_2 : t_A + 2L_s - j \leq t_B \leq t_s - 1$

$\text{Case}_3 : t_A - L_s + 1 \leq t_B \leq t_A$

$\text{Case}_4 : t_A + 1 \leq t_B \leq t_A + L_s - 1$

$\text{Case}_5 : t_A + L_s \leq t_B \leq t_A + 2L_s - j - 1.$

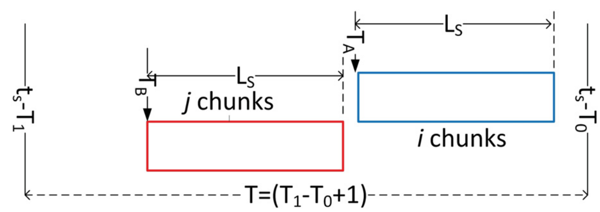
We, now calculate the probability for each case, such that peer A will not be interested in the chunks of its neighbor B.

Let $f_k(i,j)$ be the probability in Case_k, where $k = 1...5$, such that, in the current time slot, peer A is not interested in the chunks of its neighbor B under the condition of $Case_k$. We assume that the playback pointer of a peer is uniformly distributed within the interval $[t_s - T_0, t_s - T_1]$ and $T = (T_1 - T_0 + 1)$. So, the probability that the playback pointer of peer A is in a given position is $1/T$. It is to be noted that, the definition of the probability of the

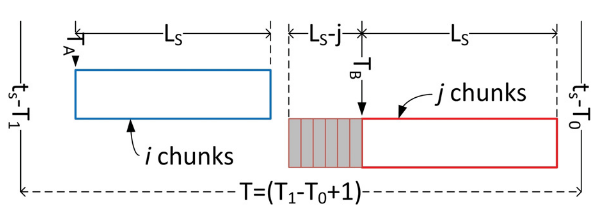
playback pointer of any peer to be in a given position will be same for all the cases and the model will work with other distributions as well, however it will be more complicated with those distributions.
As the $f_k(i,j)$ probabilities are mutually exclusive:

$$f(i,j) \quad = \quad f_1(i,j) \quad + \quad f_2(i,j) \quad + \quad f_3(i,j) \quad + \quad f_4(i,j) \quad + \quad f_5(i,j). \quad (65)$$

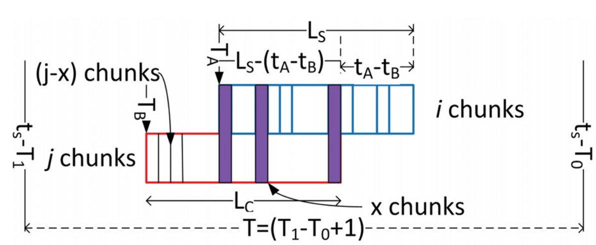
Computations of the $f_k(i,j)$ probabilities are as follows.
$Case_1 : t_s - T_1 \leq t_B \leq t_A - L_s$ (see Figure 21(a)):



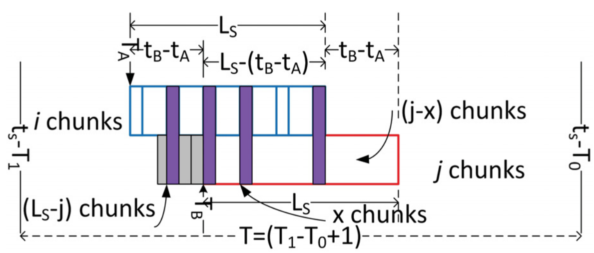
(a) $Case_1$



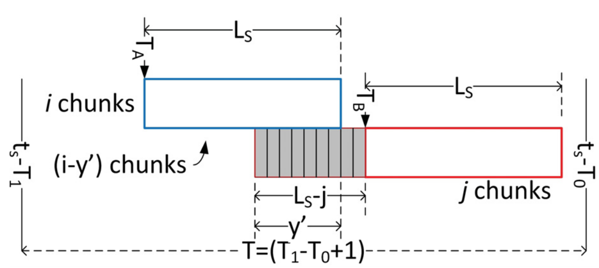
(b) $Case_2$



(c) $Case_3$



(d) $Case_4$



(e) $Case_5$

Figure 21: Overlapping of i useful streaming chunks of peer A with j useful chunks of one its random neighbor B for 5 mutually exclusive cases

Peer A has i useful chunks and its random neighbor B has j useful chunks in their respective buffers. Moreover, those buffers have no overlapped content. In this case, peer A has already played all the j useful chunks of B and has no further interest in them. It follows:

$$f_1(i,j) = \sum_{t_A=t_s-T_1+L_s}^{t_s-T_0} \sum_{t_B=t_s-T_1}^{t_A-L_s} \frac{1}{T^2}. \quad (66)$$

In this case, probability that A is not interested in the chunks of $B = 1$. Here, probability that the playback pointer, t_A, of peer A is at a given position is $1/T$ and the probability that the playback pointer t_B, of peer B is at a given position is also $1/T$.

$Case_2 : t_A + 2L_s - j \le t_B \le t_s - T_0$ (see Figure 21(b)):

In this case, buffers of peer A and neighbor B have no content overlap and as opposed to $Case_1$, buffer of the neighbor B contains more recent pieces than peer A. The lowest sequence number of the $(L_s - j)$ old chunks (residing in the buffer) of peer B is even higher than the highest sequence number of the chunks that A can store in its buffer in the current time slot. Hence, peer A has no interest in its neighbor B. It follows:

$$f_2(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0-(2L_s-j)} \sum_{t_B=t_A+2L_s-j}^{t_s-T_0} \frac{1}{T^2}. \quad (67)$$

$Case_3[t_A - L_s + 1 \le t_B \le t_A]$ (see Figure 21(c)):

Among the j useful chunks of neighbor B, x chunks are already present in the overlapping portion of the buffer of peer A. If A is not interested in B, then whatever chunks neighbor B has in the overlapping portion, peer A already has those chunks. The remaining $(j - x)$ chunks of B are of no interest to A (as those chunks are already played by A). It follows:

$$f_3(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_B=\max\{t_s-T_1,t_A-L_s+1\}}^{t_A} \sum_{x=\max\{0,j-(t_A-t_B)\}}^{\min\{L_s-(t_A-t_B),i,j\}} \left[\left(\frac{\binom{t_A-t_B}{j-x}\binom{L_s-(t_A-t_B)}{x}}{\binom{L_s}{j}}\right)\left(\frac{\binom{L_s-x}{i-x}}{\binom{L_s}{i}}\right)\right]\frac{1}{T^2}. \quad (68)$$

In this case, probability that A is not interested in the chunks of B is computed inside the square braces. The first part inside the square braces denotes the probability that neighbor B has j chunks in its buffer and among these j chunks, x chunks are in the portion (i.e., $L_s - (t_A - t_B)$) which is overlapped with the buffer of peer A. Remaining $j - x$ chunks of peer B lies in the non-overlapped region, $t_A - t_B$. The second part denotes that peer A has i chunks where these i chunks include all the x chunks which are in the buffer of neighbor B.

$Case_4 : t_A + 1 \le t_B \le t_A + L_s - 1$ (see Figure 21(d)):

Neighbor B has x (out of j) useful chunks in the overlapping portion. If peer A is not interested in B then A also has all the x chunks in its buffer. Moreover, peer A has all the $(L_s - j)$ old chunks of B. The remaining $(j - x)$ chunks of B are of no interest to A (as those chunks are too new for it for the current time slot). We then have:

$$f_4(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_B=\min\{t_s-T_0,t_A+1\}}^{\min\{t_s-T_0,t_A+L_s-1\}} \sum_{x=\max\{0,j-(t_B-t_A)\}}^{\min\{L_s-(t_B-t_A),i-\min\{(t_B-t_A),(L_s-j)\},j\}} \left[\left(\frac{\binom{t_B-t_A}{j-x}\binom{L_s-(t_B-t_A)}{x}}{\binom{L_s}{j}}\right)\left(\frac{\binom{L_s-y}{i-y}}{\binom{L_s}{i}}\right)\right]\frac{1}{T^2}. \quad (69)$$

where $y = x + \min\{(t_B - t_A), (L_s - j)\}$.
Here, probability that A is not interested in the chunks of B is computed inside the square braces where the first part inside the square braces denotes the probability that neighbor B has j chunks in its buffer and among these j chunks, x chunks are in the portion, $L_s-(t_B-t_A)$, which is overlapped with the buffer of peer A. Remaining $j - x$ useful chunks lies in the non-overlapped region, $t_B - t_A$, of peer B. However, in this case, B has $L_s - j$ old chunks which are overlapped with the first $t_B - t_A$ portion of the buffer of peer A The second part denotes that peer A has i chunks where these i chunks include all the x useful chunks and all the $L_s - j$ old chunks of the buffer of neighbor B (i.e., defined under y).
$Case_5 : t_A + L_s \leq t_B \leq t_A + 2L_s - j - 1]$(see Figure 21(e)):
In this case, there is a overlapping between the i useful chunks of peer A and y' (among $(L_s - j)$) old chunks of neighbor B. Peer A is not interested in the j useful chunks of neighbor B (as those are too new for A in the current time slot). We have:

$$f_5(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0-L_s} \sum_{t_B=t_A+L_s}^{\min\{t_s-T_0,[(t_A+L_s)+L_s-j-1]\}} \left[\frac{\binom{L_s-y'}{i-y'}}{\binom{L_s}{i}}\right]\frac{1}{T^2}. \quad (70)$$

where $y' = t_A + L_s - [t_B - (L_s - j)]$.
We recall the definition of U_i and express as follows:

$$U_i = \sum_{j=0}^{L_s} (1 - f(i,j)) \times P_j. \quad (71)$$

where, P_j is the probability that a random peer in the network has j useful chunks. Next, we calculate the probabilities of downloading chunks.

3.5.1 Probabilities of Requesting and Downloading Chunks

We recall that, any random peer in this network has H fixed number of neighbors and a buffer L_s of fixed length. It is also assumed that at any time slot if any randomly picked peer (e.g. peer A) receives more than one request, u of them (including the one from A) will be fulfilled randomly by peer B. The maximum number of requests that an arbitrary peer (e.g., A) having i useful chunks in the current time slot can send to its neighbors: $\min\{H, i'\}$ where, $i' = (L_s - i)$ is maximum number of chunks that peer A wants to download at that specific time slot and total number of neighbors $H = (H_c + H_f)$.
Next, we define $F(H, i', k)$, as the probability that a randomly selected peer which has i useful chunks in its buffer, in a specific time slot, looking for $i' = (L_s - i)$, chunks in that time slot, sends k requests to its neighbors.
$F(H, i', k)$ can be recursively calculated as follows:

$$F(H, i', k) = U_i F(h - 1, i' - 1, k - 1) + (1 - U_i) F(h - 1, i', k) \quad (72)$$

where, $i' = (L_s - i)$, $H \geq h \geq 0$, $\min\{i', h\} \geq k \geq 0$ and $L_s \geq i \geq 0$,
The first component of the right hand side in (72) assumes the peer receives one of the requested chunks from one of the H viewing neighbors. Hence, in the second part, it is looking for one less chunk (i.e., $i' - 1$) from the remaining neighbors (i.e., $h - 1$) and sends one less request (i.e., $k - 1$) to them.
The recurrence relation (72) has the following initial conditions:

$$F(H, i', k) = \begin{cases} 0 & \text{if } (h = 0 \text{ or } i' = 0) \text{ and } k > 0 \\ 1 & \text{if } (h = 0 \text{ or } i' = 0) \text{ and } k = 0. \end{cases}$$

Next, we calculate (i) the expected-value of $F(h, i', k)$ as $\overline{L}_i$ and (ii) define $\overline{k}$ as the average number of requests that any arbitrary peer sends to its neighbors at any given time slot.

$$\overline{L}_i = \sum_{k=0}^{\min\{i', H\}} kF(h, i', k); \quad (73)$$

$$\overline{k} = \sum_{i=0}^{L_s} P_i \overline{L}_i. \quad (74)$$

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

$$\overline{X} = \frac{(H-1)\overline{k}}{H} \tag{75}$$

This shows that $H-1$ neighbors send $H\overline{k}$ requests to their neighbors and on an average each of the H neighbors receive $H\overline{k}/H$ requests from them.

3.5.2 Probabilities of Fulfilling Requests by the Contributing Peers and the Free riding Peers

We recall that a peer may have both contributing and Free riding neighbors at the same time. Next, we define the following:
(i) Q^{c2n} as the probability that a randomly selected contributing peer (e.g., peer B) arbitrarily fulfills u requests (i.e., including the one from peer A) among all the requests it received from its neighbors:

$$Q^{c2n} = \frac{u}{1+\overline{\overline{X}}} \tag{76}$$

Similarly, we define:
(ii) Q^{f2n} as the probability that a randomly selected Free riding peer (e.g., peer D) arbitrarily fulfills u requests (i.e., including the one from peer A) among all the requests it received from its neighbors:

$$Q^{f2n} = \frac{u}{1+\overline{\overline{X}}} \times q_f \tag{77}$$

Now, we compute the transition probability in the discrete-time stochastic model shown in Figure 18 as follows:
We define, $r_{i,n}$, probability that a random peer having i useful chunks at a given time slot, downloads n chunks in the same time slot. Hence, depending on the type of neighbors (i.e., contributing and Free riding), $r_{i,n}$ can be expressed as follows:

$$r_{i,n} = \sum_{k=n}^{\min\{L_s-i,H\}} \sum_{k_f=0}^{k} F(H,L_s-i,k) \left[\binom{k}{k_f}(P^f)^{k_f}(1-P^f)^{(k-k_f)}\right]$$
$$\sum_{l=0}^{\min\{n,k_f\}} \left[\binom{k_f}{l}(Q^{f2n})^{l}(1-Q^{f2n})^{(k_f-l)}\right]\left[\binom{k-k_f}{n-l}(Q^{c2n})^{(n-l)}(1-Q^{c2n})^{(k-k_f-n+l)}\right] \tag{78}$$

Next, we define, M_i, probability that a randomly picked peer in the network with the buffer L_s and i distributed useful chunks, in its buffer in a specific time slot, can play a chunk in the same time slot. Since we do not set priority to any of the chunks, the probability that we have as follows:

$$M_i = \frac{i}{L_s} \qquad i = 0, 1, ..., L_s. \tag{79}$$

If the system is in state P_i, the probability that the system jumps to state P_{i+k} in the next time slot is given by:

$$B_{i,i+k} = (1 - M_i)r_{i,k} + M_i r_{i,k+1} \tag{80}$$

The first term on the right hand side corresponds to a situation where the first chunk is missing (i.e., the peer cannot playback) and k chunks are downloaded. The second term corresponds to a situation where the peer plays the first chunk and downloads $(k+1)$ chunks in the current time slot. When the system is in steady state, the probability distribution of any state in the model (shown in Figure 18) does not change with time and we get the following relation:

$$B_{i+1,i} \times P_{i+1} + \left(\sum_{k=0}^{\min\{H,i\}} (B_{i-k,i} \times P_{i-k}) \right) - \sum_{k=1}^{L_s - i} (B_{i,i+k} \times P_i) - B_{i,i-1} \times P_i = 0. \tag{81}$$

It is difficult to find a closed form of the solution of equation (81) in order to get the peer distribution $\{P_i\}$. This is because, in the stochastic model, we have $L_s + 1$ unknown states and $L_s + 1$equation for the solution. However, it is possible to solve (31) numerically and get results very close to the close form of solution.

Next, we define P_{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 chunk at this time slot. It can be expressed as follows:

$$P_{cont} = \sum_{i=0}^{L_s} P_i \times M_i. \tag{82}$$

3.6 A Simple Incentive for the Peers to Cooperate

In this section we designed a simple incentive mechanism and assume that this mechanism is present in the system to help the peers (specially Free riders) to cooperate each other in a P2P Live streaming system. In the previous model presented in Section 3.3, there was no incentive mechanism present and no monitoring was done to determine how generously a peer is granting a request coming from its neighbor. We propose a simplified incentive mechanism in order to make it useful for our analysis. Later on, we can use this analysis for designing more efficient incentive mechanisms for specific cases.

In the proposed incentive mechanism, we assume that each peer maintains a rank R_j for its neighbors j: So, the rank R_j of a viewing neighbor j is defined as the ratio of contribution made by n_j (i.e., n_j) to the contribution sought from n_j (i.e., N_j):

$$R_j = \frac{n_j}{1 + N_j} \tag{83}$$

where, N_j represents the number of times neighbor j was asked to upload streaming content and n_j represents the number of times neighbor j has responded positively to those N_j upload request of the current peer. In the denominator, 1 is added in order to avoid 'divide by zero' at the beginning. After a while the effect of 1 in the denominator will be very insignificant to consider.

This simple incentive mechanism may not be the best choice for a real life P2P live streaming system. However, the main purpose of the proposed incentive mechanism is to proceed with feasible theoretical analysis. Next, we develop our stochastic models for P2P Live streaming systems having such an incentive mechanism available.

3.7 Analysis of the Stochastic Model with Incentive

In this model we assume that a peer can be identified either as a contributing neighbor or a Free riding peer.

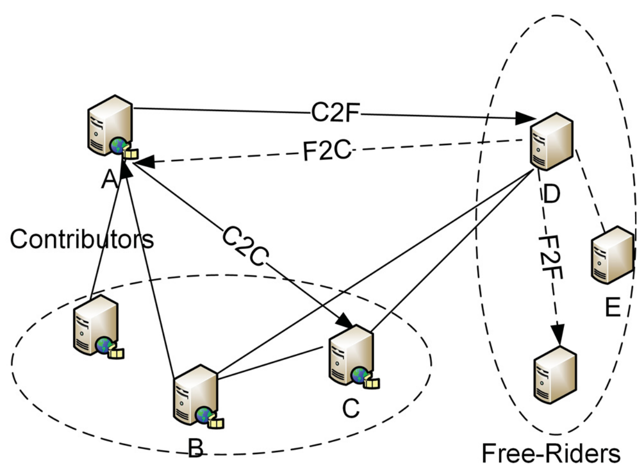


Figure 22: A Peer having 4 different types of interactions

As illustrated in Figure 22, we identify the following 4 interaction scenarios for a peer in this model:

- **Contributing-to-Contributing (C2C) Scenario:** In this scenario, a contributing peer requests and downloads from its contributing neighbor.

- **Contributing-to-FreeRiding (C2F) Scenario:** In this scenario, a contributing peer requests and attempts to download from its Free riding neighbor.
- **FreeRiding-to-Contributing (F2C) Scenario:** In this scenario, a Free riding peer requests and downloads from its contributing neighbor.
- **FreeRiding-to-FreeRiding (F2F) Scenario:** In this scenario, a Free riding peer requests and attempts to download from its Free riding neighbor.

We analyze the stochastic model and calculate the probabilities for appropriate scenarios, C2C, C2F, F2C and F2F. Next we calculate this probability distribution for the $C2C$ interaction.

3.7.1 Contributing-to-Contributing (C2C) Peers' Interactions

In the C2C scenario, consider an arbitrary peer (e.g., peer A) and one of its neighbors (e.g., peer B) such that both are with playback buffer of equal size, L_s. Suppose, at the current time slot, peer A has i useful chunks and neighbor B has j useful chunks in their respective buffers. t_A and t_B are the sequence numbers of the chunks to be played by the corresponding peer A and its neighbor B. Based on that, we calculate, U_i^{c2c}, the probability that peer A, while having i useful chunks in its buffer in a specific time slot, is interested in getting a chunk from a random neighbor, B. In order to calculate, U_i^{c2c}, let $f^{c2c}(i,j)$ be the probability that, at a given time slot, peer A with i useful chunks in its buffer is not interested in any of the j useful chunks of B. We then have:

$$U_i^{c2c} = \sum_{j=0}^{L_s} \left(1 - f^{c2c}(i,j)\right) P_j^{c2c}. \tag{84}$$

where P_j^{c2c} is the probability that a random contributing peer in the network has j useful chunks. We calculate $f^{c2c}(i,j)$ for each of the following five mutually exclusive cases:
$\text{Case}_1^{c2c} : t_s - T_1 \leq t_B \leq t_A - L_s$
$\text{Case}_2^{c2c} : t_A + 2L_s - j \leq t_B \leq t_s - T_0$
$\text{Case}_3^{c2c} : t_A - L_s + 1 \leq t_B \leq t_A$
$\text{Case}_4^{c2c} : t_A + 1 \leq t_B \leq t_A + L_s - 1$
$\text{Case}_5^{c2c} : t_A + L_s \leq t_B \leq t_A + 2L_s - j - 1.$

We denote by $f_k^{c2c}(i,j)$ be the probability that peer A is not interested in the useful pieces of peer B and these two peers are in Case_k^{c2c}, where $k = 1...5$. We assume that the playback pointer of a peer is uniformly distributed in interval $[t_s - T_0, t_s - T_1]$ and $T = (T_1 - T_0 + 1)$.

So, the probability that the playback pointer of peer A is in a given position is $1/T$. It is to be noted that, the definition of the probability of the playback pointer of any peer to be in a given position will be same for all the cases and the model will work with other distributions as well, however it will be more complicated

As the $f_k^{c2c}(i,j)$ probabilities are mutually exclusive:

$$f^{c2c}(i,j) \quad = \quad f_1^{c2c}(i,j) \ + \ f_2^{c2c}(i,j) \ + \ f_3^{c2c}(i,j) \ + \ f_4^{c2c}(i,j) \ + \ f_5^{c2c}(i,j). \quad (85)$$

Computations of the $f_k^{c2c}(i,j)$ probabilities are as follows.

$Case_1^{c2c} : t_s - T_1 \leq t_B \leq t_A - L_s$ (see Figure 21(a)):

Peer A has i useful chunks and its random neighbor B has j useful chunks in their respective buffers. Moreover, those buffers have no content overlap. Peer A has already played all the j chunks of B and has no further interest in them. We suppose $\overline{f}_1^{c2c}(i,j)$, be the probability that A is not interested in B (i.e., irrespective of it the interval, T). We assume that the playback pointer of a peer is uniformly distributed in interval $[t_s - T_0, t_s - T_1]$ and $T = (T_1 - T_0 + 1)$. So, the probability that the playback pointer of peer A is in a given position is $1/T$. It is to be noted that, the model will also work with other distributions, however it will be more complicated

Under the same assumption, the definition of the probability of the playback pointer of any peer to be in a given position will be same for all the subsequent cases. It follows:

$$f_1^{c2c}(i,j) = \sum_{t_A = t_s - T_1 + L_s}^{t_s - T_0} \ \sum_{t_B = t_s - T_1}^{t_A - L_s} \frac{1}{T^2}. \quad (86)$$

In this case, $\overline{f}_1^{c2c}(i,j)$=1

$Case_2^{c2c} : t_A + 2L_s - j \leq t_B \leq t_s - T_0$ (see Figure 21(b)):

In this case, buffers of peer A and neighbor B have no content overlap and as opposed to $Case_1^{c2c}$, buffer of the neighbor B contains more recent pieces than peer A. The lowest sequence number of the $(L_s - j)$ old chunks (residing in the buffer) of peer B is even higher than the highest sequence number of the chunks that A can store in its buffer in the current time slot. Hence, peer A has no interest in its neighbor B. It follows:

$$f_2^{c2c}(i,j) = \sum_{t_A = t_s - T_1}^{t_s - T_0 - (2L_s - j)} \ \sum_{t_B = t_A + 2L_s - j}^{t_s - T_0} \frac{1}{T^2}. \quad (87)$$

In this case, $\overline{f}_2^{c2c}(i,j)$=1

$Case_3^{c2c}[t_A - L_s + 1 \leq t_B \leq t_A]$ (see Figure 21(c)):

Among the j useful chunks of neighbor B, x chunks are already present in the overlapping portion of the buffer of peer A. If A is not interested in B, then whatever chunks neighbor B has in the overlapping portion, peer A already has those chunks. The remaining $(j-x)$ chunks of B are of no interest to A (as those chunks are already played by A). It follows:

$$f_3^{c2c}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_B=\max\{t_s-T_1,t_A-L_s+1\}}^{t_A} \sum_{x=\max\{0,j-(t_A-t_B)\}}^{\min\{L_s-(t_A-t_B),i,j\}} \left[\left(\frac{\binom{t_A-t_B}{j-x}\binom{L_s-(t_A-t_B)}{x}}{\binom{L_s}{j}}\right)\left(\frac{\binom{L_s-x}{i-x}}{\binom{L_s}{i}}\right)\right]\frac{1}{T^2}. \quad (88)$$

In this case, $\overline{f}_3^{c2c}(i,j)$ is calculated inside the square braces. The first part of $\overline{f}_3^{c2c}(i,j)$ denotes the probability that neighbor B has j chunks in its buffer and among these j chunks, x chunks are in the portion which is overlapped with the buffer of peer A. The second part denotes that peer A has i chunks where these i chunks include all the x chunks which are in the buffer of neighbor B. Under the same assumption, $\overline{f}_k^{c2c}(i,j)$ will be calculated in the same manner for all the subsequent cases.

$Case_4^{c2c}: t_A+1 \le t_B \le t_A+L_s-1$ (see Figure 21(d)):

Neighbor B has x (out of j) useful chunks in the overlapping portion, however, peer A already has all the x chunks in its buffer. Moreover, peer A has all the (L_s-j) old chunks of B. The remaining $(j-x)$ chunks of B are of no interest to A (as those chunks are too new for it for the current time slot). We then have:

$$f_4^{c2c}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_B=\min\{t_s-T_0,t_A+1\}}^{\min\{t_s-T_0,t_A+L_s-1\}} \sum_{x=\max\{0,j-(t_B-t_A)\}}^{\min\{L_s-(t_B-t_A),i-\min\{(t_B-t_A),(L_s-j)\},j\}} \left[\left(\frac{\binom{t_B-t_A}{j-x}\binom{L_s-(t_B-t_A)}{x}}{\binom{L_s}{j}}\right)\left(\frac{\binom{L_s-y}{i-y}}{\binom{L_s}{i}}\right)\right]\frac{1}{T^2}. \quad (89)$$

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

$Case_5^{c2c}: t_A+L_s \le t_B \le t_A+2L_s-j-1]$(see Figure 21(e)):

In this case, there is a overlapping between the i useful chunks of peer A and y' (among (L_s-j)) old chunks of neighbor B. Peer A is not interested in the j useful chunks of neighbor B (as those are too new for A in the current time slot). We have:

$$f_5^{c2c}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0-L_s} \sum_{t_B=t_A+L_s}^{\min\{t_s-T_0,[(t_A+2L_s)-j-1]\}} \left[\frac{\binom{L_s-y'}{i-y'}}{\binom{L_s}{i}}\right]\frac{1}{T^2}. \quad (90)$$

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

3.7.2 Contributing-to-Free-riding (C2F) Peers' Interactions

In the c2f scenario, consider an arbitrary peer (e.g., peer A with playback buffer L_s) and one of its Free riding neighbors (e.g., peer D with playback buffer L_s). Suppose, at the current time slot, peer A has i useful chunks and neighbor D has j useful chunks in their respective buffers. Based on that, we will calculate, U_i^{c2f}, probability that peer A having i useful chunks in its buffer in a specific time slot, is interested to get a chunk from a random cooperative neighbor D. In order to calculate, U_i^{c2f}, let $f^{c2f}(i,j)$ be the probability that, at a given time slot, peer A with i useful chunks in its buffer is not interested in any of the j useful chunks of D. We then have:

$$U_i^{c2f} = \sum_{j=0}^{L_s} \left(1 - f^{c2f}(i,j)\right) P_j^{c2f}. \tag{91}$$

where P_j^{c2f} is the probability that a random peer in the network has j useful chunks.

We denote by $f_k^{c2f}(i,j)$ the $f^{c2f}(i,j)$ probability in Case_k^{c2f}, where $k = 1...5$. As the $f_k^{c2f}(i,j)$ probabilities are mutually exclusive:

$$f^{c2f}(i,j) = f_1^{c2f}(i,j) + f_2^{c2f}(i,j) + f_3^{c2f}(i,j) + f_4^{c2f}(i,j) + f_5^{c2f}(i,j). \tag{92}$$

Computations of the $f_k^{c2f}(i,j)$ probabilities are as follows.
$\text{Case}_1^{c2f} : t_s - T_1 \leq t_D \leq t_A - L_s$
$\text{Case}_2^{c2f} : t_A + 2L_s - j \leq t_D \leq t_s - T_0$
$\text{Case}_3^{c2f} : t_A - L_s + 1 \leq t_D \leq t_A$
$\text{Case}_4^{c2f} : t_A + 1 \leq t_D \leq t_A + L_s - 1$
$\text{Case}_5^{c2f} : t_A + L_s \leq t_D \leq t_A + 2L_s - j - 1$.
We, now calculate the probability, that peer A will not be interested in the chunks of neighbor D.
Let $f_k^{c2f}(i,j)$ be the probability in Case_k^{c2f}, where $k = 1...5$, such that, in the current time slot, peer A is not interested in the streaming chunks of neighbor D under the condition of $Case_k^{c2f}$.
We Calculate these probability values of C2F for 5 cases in a similar way those are done in Section 3.7.1.

3.7.3 Free-riding-to-Contributing (F2C) Peers' Interactions

In the f2c scenario, consider an arbitrary Free riding peer (e.g., peer D with playback buffer L_s) and one of its contributing neighbors (e.g., peer A with playback buffer L_S) such that they are viewers of two different channels. Suppose, at the current time slot, peer D has i useful chunks and its neighbor A has j useful chunks in their respective buffers. Based on that, we will calculate, U_i^{f2c}, probability that peer D having i useful chunks in its buffer in a specific time slot, is interested to get a chunk from a random contributing neighbor A. In order to calculate, U_i^{f2c}, let $f^{f2c}(i,j)$ be the probability that, at a given time slot, peer D with i useful chunks in its buffer is not interested in any of the j useful chunks of contributing neighbor A. We then have:

$$U_i^{f2c} = \sum_{j=0}^{L_s} \left(1 - f^{f2c}(i,j)\right) P_j^{f2c}. \tag{93}$$

where P_j^{f2c} is the probability that a random peer in the network has j useful chunks.

We denote by $f_k^{f2c}(i,j)$ the $f^{f2c}(i,j)$ probability in Case$_k^{f2c}$, where $k = 1...5$. As the $f_k^{f2c}(i,j)$ probabilities are mutually exclusive:

$$f^{f2c}(i,j) = f_1^{f2c}(i,j) + f_2^{f2c}(i,j) + f_3^{f2c}(i,j) + f_4^{f2c}(i,j) + f_5^{f2c}(i,j). \tag{94}$$

Computations of the $f_k^{f2c}(i,j)$ probabilities are as follows.

$Case_1^{f2c} : t_s - T_1 \leq t_A \leq t_D - L_s$

$Case_2^{f2c} : t_D + 2L_s - j \leq t_A \leq t_s - T_0$

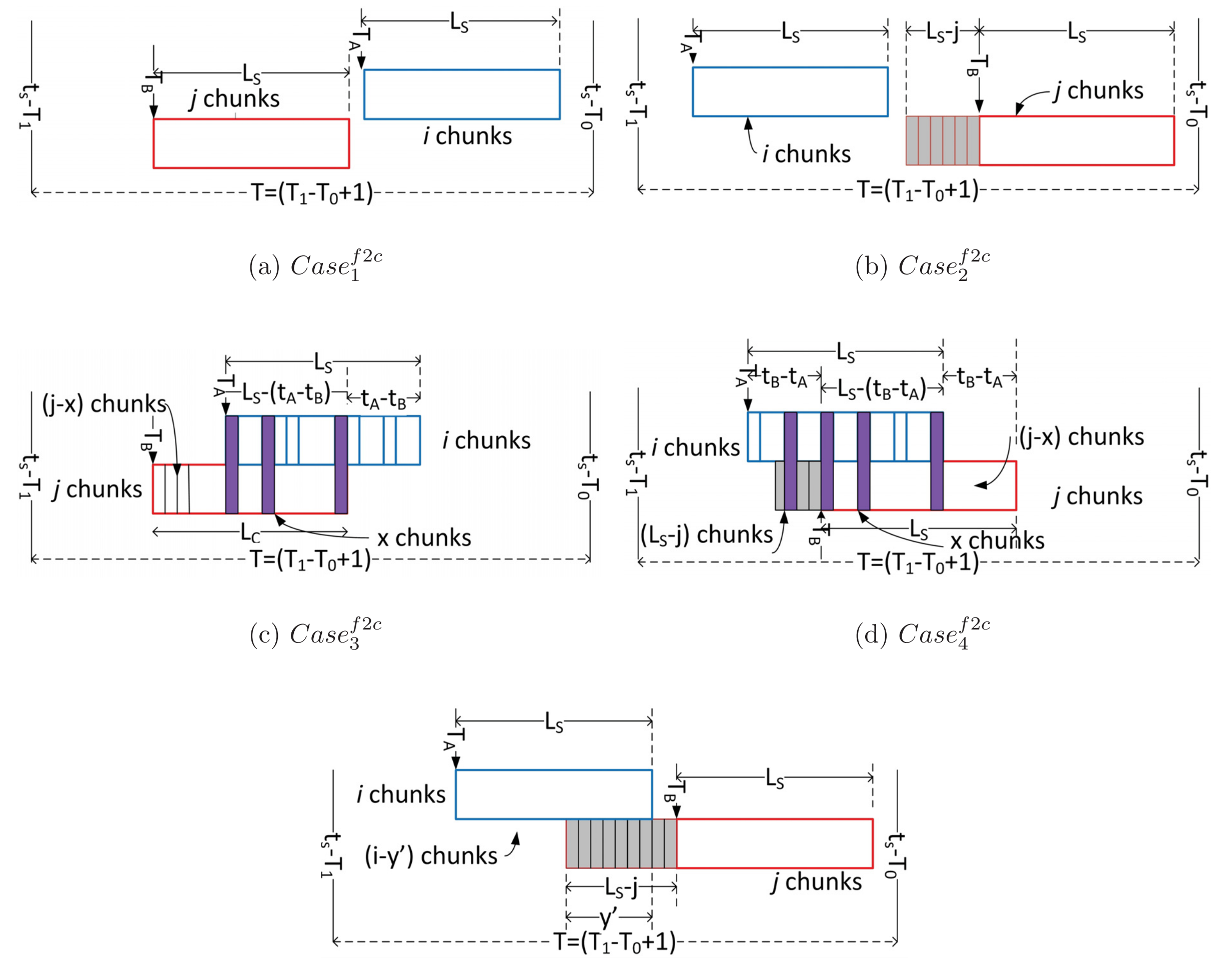
$Case_3^{f2c} : t_D - L_s + 1 \leq t_A \leq t_D$

$Case_4^{f2c} : t_D + 1 \leq t_A \leq t_D + L_s - 1$

$Case_5^{f2c} : t_D + L_s \leq t_A \leq t_D + 2L_s - j - 1$

We, now calculate the probability, that Free riding peer D will not be interested in the chunks of its neighbor A. Let $f_k^{f2c}(i,j)$ be the probability for Case$_k^{f2c}$, where $k = 1...5$, such that, in the current time slot, Free riding peer D is not interested in the streaming chunks of its neighbor A under the condition of $Case_k^{f2c}$.

We Calculate these probability values of F2C for 5 cases in a similar way those are done in Section 3.7.1. Figure 24 shows different states of A discrete-time Stochastic model for an arbitrarily selected Free riding peer in the system.



(a) $Case_1^{f2c}$ (b) $Case_2^{f2c}$

(c) $Case_3^{f2c}$ (d) $Case_4^{f2c}$

(e) $Case_5^{f2c}$

Figure 23: Overlapping streaming chunks of Free riding peer D and one of its contributing neighbor A, in the Cooperative-to-contributing (f2c) cases

3.7.4 Free riding-to-Free riding (F2F) Peers' Interactions

In the F2F scenario, consider an arbitrary Free riding peer (e.g., peer D) and one of its neighbors (e.g., peer E) such that both with playback buffer of equal size, L_s. Suppose, at the current time slot, peer D has i useful chunks and neighbor E has j useful chunks in their respective buffers. t_D and t_E are the sequence numbers of the chunks to be played by the corresponding peer D and its neighbor E. Based on that, we calculate, U_i^{f2f}, the probability that peer D, while having i useful chunks in its buffer in a specific time slot, is interested in getting a chunk from a random Free riding neighbor, E. In order to calculate, U_i^{f2f}, let $f^{f2f}(i,j)$ be the probability that, at a given time slot, peer D with i useful chunks in its

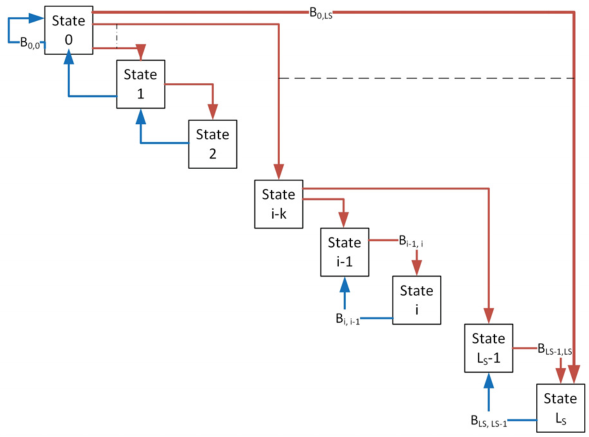


Figure 24: A discrete-time stochastic model for Free riding peers in a P2P Live streaming system having buffer size L_s.

buffer is not interested in any of the j useful chunks of E. We then have:

$$U_i^{f2f} = \sum_{j=0}^{L_s} \left(1 - f^{f2f}(i,j)\right) P_j^{f2f}. \tag{95}$$

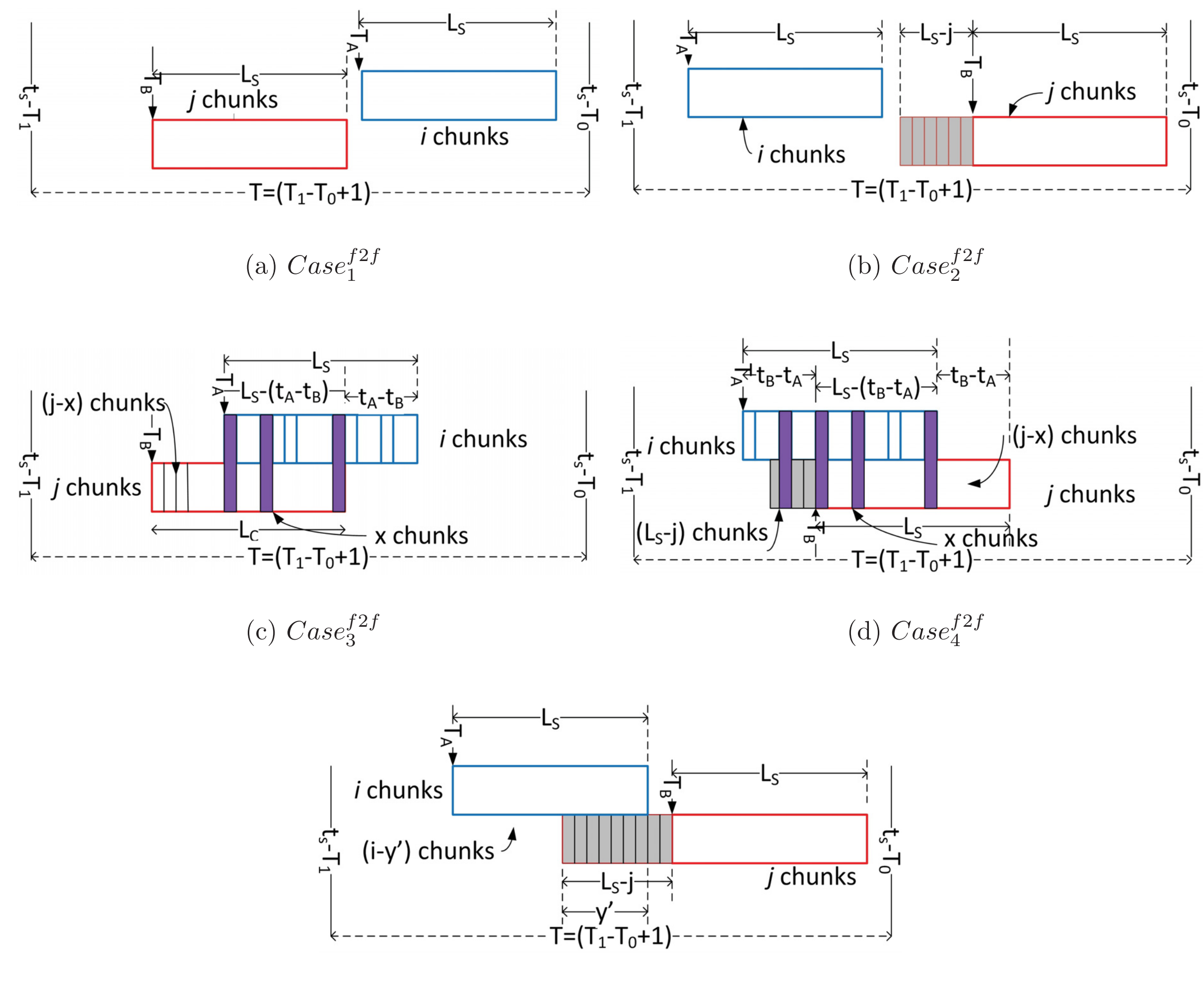
where P_j^{f2f} is the probability that a random Free riding peer in the network has j useful chunks. We calculate $f^{f2f}(i,j)$ for each of the following five mutually exclusive cases:
Case_1^{f2f} : $t_s - T_1 \leq t_B \leq t_A - L_s$
Case_2^{f2f} : $t_A + 2L_s - j \leq t_B \leq t_s - T_0$
Case_3^{f2f} : $t_A - L_s + 1 \leq t_B \leq t_A$
Case_4^{f2f} : $t_A + 1 \leq t_B \leq t_A + L_s - 1$
Case_5^{f2f} : $t_A + L_s \leq t_B \leq t_A + 2L_s - j - 1$. We denote by $f_k^{f2f}(i,j)$ be the probability that peer D is not interested in the useful pieces of peer E and these two peers are in Case_k^{f2f}, where $k = 1...5$. We assume that the playback pointer of a peer is uniformly distributed within the interval $[t_s - T_0, t_s - T_1]$ and $T = (T_1 - T_0 + 1)$. So, the probability that the playback pointer of peer D is in a given position is $1/T$. It is to be noted that, the definition of the probability of the playback pointer of any peer to be in a given position will be same for all the cases and the model will work with other distributions as well, however it will be more complicated
It follows:

$$f_k^{f2f}(i,j) = \overline{f}_k^{f2f}(i,j) \cdot \frac{1}{T^2}. \tag{96}$$

Under these assumption, the definition of $f_k^{f2f}(i,j)$ will be same for all the cases in Case$_k^{f2f}$, where $k = 1...5$. As the $f_k^{f2f}(i,j)$ probabilities are mutually exclusive:

$$f^{f2f}(i,j) = f_1^{f2f}(i,j) + f_2^{f2f}(i,j) + f_3^{f2f}(i,j) + f_4^{f2f}(i,j) + f_5^{f2f}(i,j). \quad (97)$$

We Calculate probability values of F2F for 5 cases in a similar way those are done in Section 3.7.1.



(a) $Case_1^{f2f}$ (b) $Case_2^{f2f}$

(c) $Case_3^{f2f}$ (d) $Case_4^{f2f}$

(e) $Case_5^{f2f}$

Figure 25: Overlapping of i useful streaming chunks of peer D with j useful chunks of one its random neighbor E in the Free riding-to-Free-riding (f2f) cases

3.7.5 Probabilities of Requesting and Downloading Chunks for a Contributing Peer with Incentive

We now assume that a peer will have both contributing and Free riding neighbors at the same time. Now, we define $F^{inc}(H_c, H_f, i', k_c, k_f)$, as the probability that a randomly selected peer (e.g., peer A) having i useful chunks in its buffer, in a specific time slot, looking for $i' = (L_s - i)$, chunks in that time slot, sends k_c requests to its H_c contributing neighbors and k_f requests to its H_f cooperative neighbors.

$F^{inc}(H_c, H_f, i', k_c, k_f)$, can be calculated from the following two recursive expressions:

$$F^{inc}(h_c, h_f, i', k_c, k_f) = U_i^{c2c} F(h_c - 1, h_f, i' - 1, k_c - 1, k_f) + (1 - U_i^{c2c}) F(h_c - 1, h_f, i', k_c, k_f) \quad (98)$$

and

$$F^{inc}(0, h_f, i'_f, 0, k_f) = U_i^{c2f} F(0, h_f - 1, i'_f - 1, 0, k_f - 1) + (1 - U_i^{c2f}) F(0, h_f - 1, i'_f, 0, k_f) \quad (99)$$

where, $i' = (L_s - i)$, $H_c \geq h_c \geq 0$, $H_f \geq h_f \geq 0$, $\min\{i', (H_c + H_f)\} \geq (k_c + k_f) \geq 0$, $\min\{i', H_c\} \geq k_c \geq 0$, $\min\{(i' - k_c), H_f\} \geq k_f \geq 0$ and $L_s \geq i \geq 0$, $i'_f = (L_s - i)$ such that $h_c = 0$ and $k_c = 0$

The first component of the right hand side in (98) assumes the peer receives one of the requested chunks from one of the h_c contributing neighbors. Hence, in the second part, it is looking for one less chunk (i.e., $i' - 1$) from the remaining contributing neighbors (i.e., $h_c - 1$) and sends one less request (i.e., $k_c - 1$). The first and the second portion of (99) can be explained similar to (98).

The recursive equation (72) has the following initial conditions:

$F^{inc}(H_c, H_f, i', k_c, k_f) =$

$$\begin{cases} 1 & \text{if } ((h_c + h_f) or i' = 0) \;\&\; k_c = k_v = 0 \\ 0 & \text{if } ((h_c + h_f) or i' = 0) \;\&\; k_c + k_v > 0 \end{cases}$$

Next, we calculate the expected-value of $F(h_c, h_f, i', k_c, k_f)$ as $\overline{L}_{i'}$ and define $\overline{k}^{inc}$ as the average number of requests that any randomly selected peer sends to its neighbors at any given time slot.

$$\overline{L}_i^{inc} = \sum_{k_c=0}^{k_c^{max}} \sum_{k_f=0}^{\min\left\{(i'-k_c),H_f\right\}} (k_c + k_f) F(H_c, H_f, i', k_c, k_f) \tag{100}$$

$$\overline{k}^{inc} = \sum_{i=0}^{L_s} (P_i \overline{L}_i^{inc}) \tag{101}$$

where, $k_c^{max} = \min\{i', H_c\}$ We define $\overline{X}^{inc}$ as the average number of requests that any randomly selected peer in the network(e.g., peer B) receives from its neighbors in addition to the request received from peer A

$$\overline{X}^{inc} = \frac{(H_c - 1 + H_f)\overline{k}^{inc}}{(H_c + H_f)} \tag{102}$$

Next, we define the followings:
(i) Q^{c2c} as the probability that a randomly selected contributing peer (e.g., peer B) arbitrarily fulfills u requests of its contributing neighbors (i.e., including the one from peer A) among all the requests it received:

$$Q^{c2c} = \frac{u}{(1 - P_f)(1 + \overline{X}^{inc}) + P_f(1 + \overline{X}^{inc})q_f} \tag{103}$$

(ii) Q^{f2c} as the probability that a randomly selected contributing peer (e.g., peer B) arbitrarily fulfills u requests for its Free riding neighbors among all the requests it received:

$$Q^{f2c} = \frac{uq_f}{(1 - P_f)(1 + \overline{X}^{inc}) + P_f(1 + \overline{X}^{inc})q_f} \tag{104}$$

(iii) Q^{c2f} as the probability that a randomly selected Free riding peer (e.g., peer D) arbitrarily fulfills u requests of its contributing neighbors(i.e., including the one from peer A) among all the requests it received:

$$Q^{c2f} = \frac{uq_f}{(1 - P_f)(1 + \overline{X}^{inc}) + P_f(1 + \overline{X}^{inc})q_f} \tag{105}$$

(iv) Q^{f2f} as the probability that a randomly selected Free riding peer (e.g., peer D) arbitrarily fulfills u requests of its Free Riding neighbors(i.e., including the one from peer A) among all the requests it received:

$$Q^{f2f} = \frac{u(q_f)^2}{(1-P_f)(1+\overline{X}^{inc}) + P_f(1+\overline{X}^{inc})q_f} \tag{106}$$

Now, we compute the probability of forward transition for a contributing peer in the discrete-time stochastic model shown in Figure 24 as follows:

We define, $r^c_{i,n}$, probability that a contributing peer having i useful chunks at a given time slot, downloads n chunks in the same time slot. Hence, depending on the type of neighbors (i.e., contributing and Free riding), $r^c_{i,n}$ can be expressed as follows:

$$r^c_{i,n} = \sum_{k=n}^{\min\{(L_s-i),(H_c+H_f)\}} \sum_{k_f=0}^{\min\{k,H_f\}} F^{inc}(H_c, H_f, (L_s-i), k_c, k_f) \sum_{l=0}^{\min\{n,k_f\}} \left[\binom{k_f}{l}(Q^{c2f})^l(1-Q^{c2f})^{(k_f-l)}\right] \left[\binom{k-k_f}{n-l}(Q^{c2c})^{(n-l)}(1-Q^{c2c})^{(k-k_f-n+l)}\right] \tag{107}$$

The first binomial distribution with parameters k_f, Q^{c2f}, $B(k_f, Q^{c2f})$ denotes that out of the k_f requests to the Free riding neighbors l of them have been granted for downloading a chunk. The second binomial distribution with parameters $(k-k_f), Q^{c2c}$, $B((k-k_f), Q^{c2c})$ denotes that out of the remaining $(k-k_f)$ requests to the contributing neighbors $(n-l)$ of them have been granted for downloading a chunk. Next, we compute the probability of forward transition for a Free rider in the discrete-time stochastic model shown in Figure 24 as follows:

We define, $r^f_{i,n}$, probability that a Free rider having i useful chunks at a given time slot, downloads n chunks in the same time slot. Hence, depending on the type of neighbors (i.e., contributing and Free riding), $r^f_{i,n}$ can be expressed as follows:

$$r^f_{i,n} = \sum_{k=n}^{\min\left\{(L_s-i),(H_c+H_f)\right\}} \sum_{k_f=0}^{\min\left\{k,H_f\right\}} F^{inc}(H_c, H_f, (L_s-i), k_c, k_f) \sum_{l=0}^{\min\left\{n,k_f\right\}} \left[\binom{k_f}{l}(Q^{f2f})^l(1-Q^{f2f})^{(k_f-l)}\right] \left[\binom{k-k_f}{n-l}(Q^{f2c})^{(n-l)}(1-Q^{f2c})^{(k-k_f-n+l)}\right] \quad (108)$$

The first binomial distribution with parameters k_f, Q^{f2f}, $B(k_f, Q^{f2f})$ denotes that out of the k_f requests to the Free riding neighbors l of them have been granted for downloading a chunk. The second binomial distribution with parameters $(k-k_f), Q^{f2c}$, $B((k-k_f), Q^{f2c})$ denotes that out of the remaining $(k-k_f)$ requests to the contributing neighbors $(n-l)$ of them have been granted for downloading a chunk. In this analysis, we assume that the desired chunks, which are scheduled to be downloaded in the current time slot, gets downloaded and stored in the buffer of that peer in the same time slot. We also assume that the streaming chunks are uniformly distributed across the buffer of a randomly selected peer in the network.
Next, we define, M^{inc}, probability that a randomly picked peer in the network with the buffer L_s and i uniformly distributed useful chunks, in its buffer in a specific time slot, can play a chunk in the same time slot. Since we do not set priority to any of the chunks, the probability that we have a chunk at the first place is:

$$M^{inc}_i = \frac{i}{L_s} \qquad i = 0, 1, ..., L_s. \quad (109)$$

If the system is in state P^{inc}_i, the probability the the system jumps to state P^{inc}_{i+k} in the next time slot is given by the probability:

$$B^{inc}_{i,i+k} = (1 - M^{inc}_i)r^{inc}_{i,k} + M^{inc}_i r^{inc}_{i,k+1} \quad (110)$$

The first term on the right hand side corresponds to the situation where the first chunk is missing (i.e., peer cannot playback in that time slot) and k chunks are downloaded. However, the second term represents a situation where the can peer play back in the current time slot and downloads $(k+1)$ chunks. When the system is in steady state, the probability distribution of any state in the model (shown in Figure 24) does not change with time and we get the following relation:

$$(B^{inc}_{i+1,i}P^{inc}_{i+1}) + \left(\sum_{k=0}^{\min\{H,i\}} (B^{inc}_{i-k,i}P^{inc}_{i-k})\right) - \left(\sum_{k=1}^{L_s-i}(B^{inc}_{i,i+k}P^{inc}_{i})\right) - (B^{inc}_{i,i-1}P^{inc}_{i}) = 0. \quad (111)$$

It is difficult to find a closed form of solution for equation 111 in order to get peer distribution $\{P^{inc}_i\}$. However, it is possible to solve (111) numerically and get result very close to the close form of solution. Next, we define P^{inc}_{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 chunk at this time slot. It can be expressed as follows:

$$P^{inc}_{cont} = \sum_{i=0}^{L_s} P^{inc}_i M^{inc}_i. \quad (112)$$

Let d^{inc}_i be the average download rate of a peer that has i useful chunks in its buffer L_s in a specific time slot:

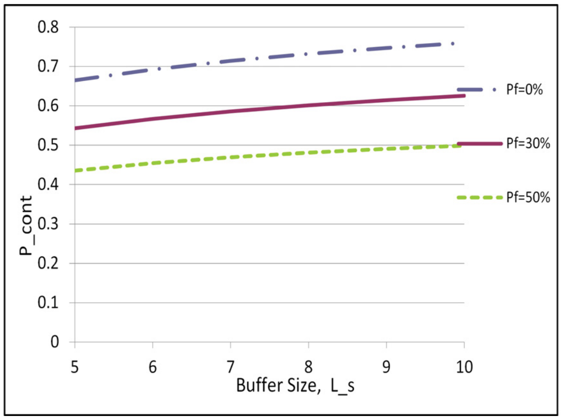
$$d^{inc}_i = \sum_{k=0}^{\min\{H,L_s-i\}} \left(k r^{inc}_{i,k}\right) \qquad i = 0, 1, ..., L_s. \quad (113)$$

Then the average download rate of a peer can be expressed as follows:

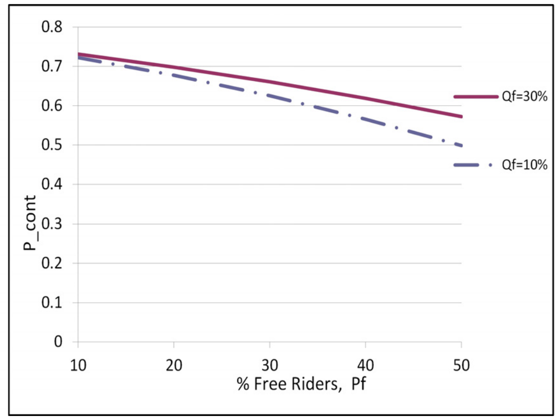
$$d^{inc} = \sum_{i=0}^{L_s} \left(d^{inc}_i P^{inc}_i\right). \quad (114)$$

3.8 Simulation and Numerical Results

In order to validate the proposed scheme, we present the numerical results (i.e., probability of continuity, P_{cont}) when different percentage of Free riding peers are present in the system. Next, we show the improvement in P_{cont} after the Free riding peers becomes more generous (i.e., with increased q_f). We assume that time is slotted and first we set the upload capacity, $u = 1$ chunk/time slot. Key parameters are summarized in Table 2. The effect of adding more Free riding peers on the probability of continuity, P_{cont} against different values of playback buffer is shown in Figure 26(a)(for $T = 15$). Therein, we find P_{cont} is affected



(a) Effect of increasing Free riders



(b) percentage of Free riders, P_f vs percentage of generosity q_f

Figure 26: Effect on probability of continuity, P_{cont}, when the system has Free riders

Parameter	minimum	Maximum
NoOf Nodes	2	15
NoOf Contributors	5	20
NoOf Free riders	1	10
Probability of Free riding, P_f	0%	50%
Probability of generosity, q_f	0%	100%
Playback Buffer, L_s	5	15
Maximum Delay, T	2	15

Table 2: Key Parameters for the Model with Free riders

significantly after percentage of Free riding peers are increased in the system. Figure 26(b), shows probability of continuity, P_{cont} against different values of percentage of Free riders, P_f. Probability of generosity, q_f are assumed fixed values. We find that the probability of continuity, P_{cont} for the peers in the system have decreased significantly.
Figure 27 shows the % Improvement in probability of continuity when the Free riding peers are acting more generously (against different values of playback buffer, L_s). We find best result for in P_{cont} under 100% generosity, q_f of the Free riders. shows the comparison between the probability of continuity, Pcont, from the simulator and Pcont derived from the stochastic model. From this, we found that the stochastic model assumes realistic assumptions. Figure 28 shows average download rate against the number of available chunks in the buffer, L_s for different percentage of Free riders. The result shows clearly higher download rate for lower percentage of Free riders in the system. *(i)* The probability of continuity is significantly dependent on the Free riders in the channel. Figure 26(a) suggests that adding more Free

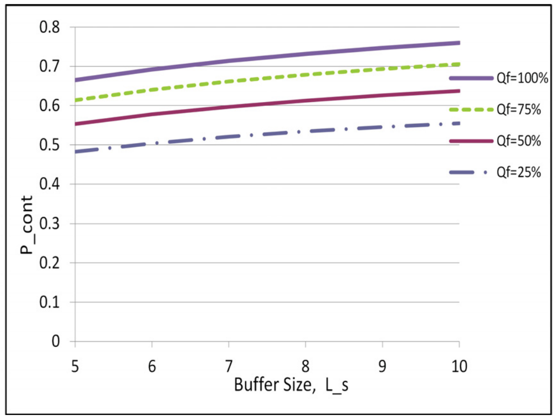


Figure 27: Effect of increased generosity, q_f, of the Free riders

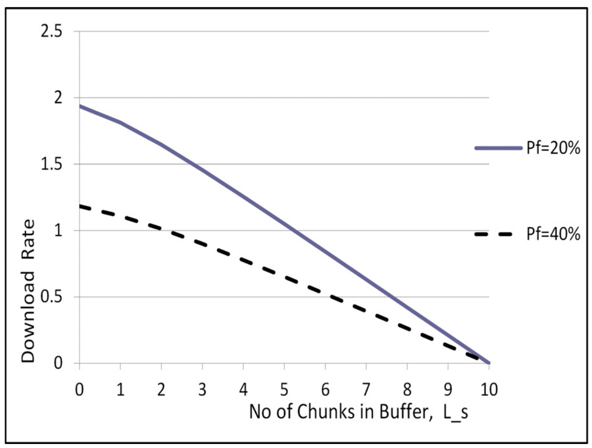


Figure 28: Average Download rate for different percentage of Free riders

riding peers obviously degrades the performance.

(ii) We also found that performance of the peers somewhat depends on the size of the playback buffer, L_s.

(iii) In the above cases, the probability of continuity is around 80%. This is mainly due to the assumption that upload capacity u is being set to 1 chunk/time slot in the model. If this condition is released, we expect the probability of continuity to be close to 1. (This is part of our future investigation).

(iv) From Figure 27, we find that increased generosity will increase the performance of all the peers in the system. Using the stochastic model we developed, we showed how probability of continuity (i.e., probability of continuous playback) can be significantly affected by Free riders. Standard compressed video (e.g, MPEG4, 480p resolution) uses bit rate of 600 kbps. In most of the residential DSL/Cable modems, available upload capacity is in the range of 800 kbps to 1.0 mbps which, in our model, is equivalent to the range of u=1.5 chunks to

u=2.0 chunks/time slot. Again, the download capacity in DSL/Cable modems is usually in the range of 5 mbps to 6 mbps which is equivalent to d=10 chunks/time slot. These justifies our assumptions in the model. Figure 29 shows comparison between the probability of continuity, P_{cont}, from the simulator and P_{cont} derived from the stochastic model. From this, we find that the stochastic model assumes realistic assumptions.

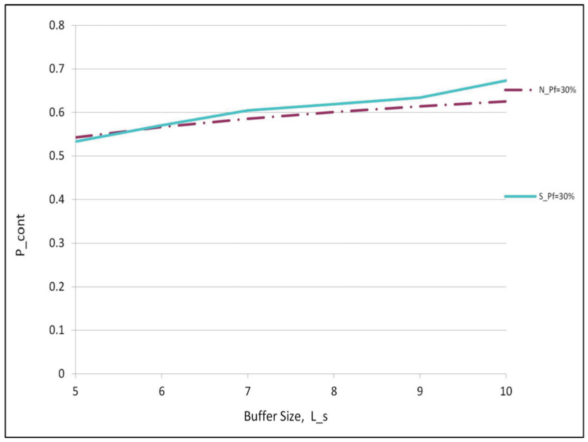


Figure 29: Comparing the Simulation result with the Numerical one

3.9 Conclusions and Future Work

We developed a simple discrete-time stochastic model for P2P Live streaming systems having Free riders. We analyzed the impacts of different parameters (e.g., probability of Free riding, probability of generosity, buffer size, maximum delay etc.) on the performance of the system. Our developed model can provide guideline for designing an efficient incentive mechanism in order to improve the performance of all the peers in the system. Future work of this topic may include: *(i)* Devising an efficient incentive mechanism for P2P Live streaming systems, assuming Free Riders present in the network. *(ii)* Developing a discrete-time stochastic model for P2P Live streaming systems in order to show the impact of the designed incentive mechanism.

Case	Probability
$f_1(i,j) =$	$$\sum_{t_A=t_s-T_1+L_s}^{t_s-T_0} \sum_{t_B=t_s-T_1}^{t_A-L_s} \frac{1}{T^2}. \quad (115)$$
$f_2(i,j) =$	$$\sum_{t_A=t_s-T_1}^{t_s-T_0-(2L_s-j)} \sum_{t_B=t_A+2L_s-j}^{t_s-T_0} \frac{1}{T^2}. \quad (116)$$
$f_3(i,j) =$	$$\sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_B=\max\{t_s-T_1,t_A-L_s+1\}}^{t_A} \sum_{x=\max\{0,j-(t_A-t_B)\}}^{\min\{L_s-(t_A-t_B),i,j\}} \left[\left(\frac{\binom{t_A-t_B}{j-x}\binom{L_s-(t_A-t_B)}{x}}{\binom{L_s}{j}}\right)\left(\frac{\binom{L_s-x}{i-x}}{\binom{L_s}{i}}\right)\right]\frac{1}{T^2}. \quad (117)$$
$f_4(i,j) =$	$$\sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_B=\min\{t_s-T_0,t_A+1\}}^{\min\{t_s-T_0,t_A+L_s-1\}} \sum_{x=\max\{0,j-(t_B-t_A)\}}^{\min\{L_s-(t_B-t_A),i-\min\{(t_B-t_A),(L_s-j)\},j\}} \left[\left(\frac{\binom{t_B-t_A}{j-x}\binom{L_s-(t_B-t_A)}{x}}{\binom{L_s}{j}}\right)\left(\frac{\binom{L_s-y}{i-y}}{\binom{L_s}{i}}\right)\right]\frac{1}{T^2}. \quad (118)$$ where $y = x + \min\{(t_B-t_A),(L_s-j)\}$.
$f_5(i,j) =$	$$\sum_{t_A=t_s-T_1}^{t_s-T_0-L_s} \sum_{t_B=t_A+L_s}^{\min\{t_s-T_0,[(t_A+L_s)+L_s-j-1]\}} \left[\frac{\binom{L_s-y'}{i-y'}}{\binom{L_s}{i}}\right]\frac{1}{T^2}. \quad (119)$$ where $y' = t_A + L_s - [t_B - (L_s - j)]$.

Table 3: Computation of Probabilities for 5 Cases without Incentive (C2N and F2N Interactions)

Case	Probability
$f_1(i,j) =$	$$f_1^{c2c}(i,j) = \sum_{t_A=t_s-T_1+L_s}^{t_s-T_0} \sum_{t_B=t_s-T_1}^{t_A-L_s} \frac{1}{T^2}. \quad (120)$$
$f_2(i,j) =$	$$\sum_{t_A=t_s-T_1}^{t_s-T_0-(2L_s-j)} \sum_{t_B=t_A+2L_s-j}^{t_s-T_0} \frac{1}{T^2}. \quad (121)$$
$f_3(i,j) =$	$$\sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_B=\max\{t_s-T_1,t_A-L_s+1\}}^{t_A} \sum_{x=\max\{0,j-(t_A-t_B)\}}^{\min\{L_s-(t_A-t_B),i,j\}} \left[\left(\frac{\binom{t_A-t_B}{j-x}\binom{L_s-(t_A-t_B)}{x}}{\binom{L_s}{j}}\right)\left(\frac{\binom{L_s-x}{i-x}}{\binom{L_s}{i}}\right)\right]\frac{1}{T^2}. \quad (122)$$
$f_4(i,j) =$	$$\sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_B=\min\{t_s-T_0,t_A+1\}}^{\min\{t_s-T_0,t_A+L_s-1\}} \sum_{x=\max\{0,j-(t_B-t_A)\}}^{\min\{L_s-(t_B-t_A),i-\min\{(t_B-t_A),(L_s-j)\},j\}} \left[\left(\frac{\binom{t_B-t_A}{j-x}\binom{L_s-(t_B-t_A)}{x}}{\binom{L_s}{j}}\right)\left(\frac{\binom{L_s-y}{i-y}}{\binom{L_s}{i}}\right)\right]\frac{1}{T^2}. \quad (123)$$ where $y = x + \min\{(t_B - t_A), (L_s - j)\}$.
$f_5(i,j) =$	$$\sum_{t_A=t_s-T_1}^{t_s-T_0-L_s} \sum_{t_B=t_A+L_s}^{\min\{t_s-T_0,[(t_A+L_s)+L_s-j-1]\}} \left[\frac{\binom{L_s-y'}{i-y'}}{\binom{L_s}{i}}\right]\frac{1}{T^2}. \quad (124)$$ where $y' = t_A + L_s - [t_B - (L_s - j)]$.

Table 4: Computation of Probabilities for 5 cases with Incentive C2C

Case	Probability
$f_1(i,j) =$	$$\sum_{t_A=t_s-T_1+L_s}^{t_s-T_0} \sum_{t_B=t_s-T_1}^{t_A-L_s} \frac{1}{T^2}. \quad (125)$$
$f_2(i,j) =$	$$\sum_{t_A=t_s-T_1}^{t_s-T_0-(2L_s-j)} \sum_{t_B=t_A+2L_s-j}^{t_s-T_0} \frac{1}{T^2}. \quad (126)$$
$f_3(i,j) =$	$$\sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_B=\max\{t_s-T_1,t_A-L_s+1\}}^{t_A} \sum_{x=\max\{0,j-(t_A-t_B)\}}^{\min\{L_s-(t_A-t_B),i,j\}} \left[\left(\frac{\binom{t_A-t_B}{j-x}\binom{L_s-(t_A-t_B)}{x}}{\binom{L_s}{j}}\right)\left(\frac{\binom{L_s-x}{i-x}}{\binom{L_s}{i}}\right)\right]\frac{1}{T^2}. \quad (127)$$
$f_4(i,j) =$	$$\sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_B=\min\{t_s-T_0,t_A+1\}}^{\min\{t_s-T_0,t_A+L_s-1\}} \sum_{x=\max\{0,j-(t_B-t_A)\}}^{\min\{L_s-(t_B-t_A),i-\min\{(t_B-t_A),(L_s-j)\},j\}} \left[\left(\frac{\binom{t_B-t_A}{j-x}\binom{L_s-(t_B-t_A)}{x}}{\binom{L_s}{j}}\right)\left(\frac{\binom{L_s-y}{i-y}}{\binom{L_s}{i}}\right)\right]\frac{1}{T^2}. \quad (128)$$ where $y = x + \min\{(t_B - t_A), (L_s - j)\}$.
$f_5(i,j) =$	$$\sum_{t_A=t_s-T_1}^{t_s-T_0-L_s} \sum_{t_B=t_A+L_s}^{\min\{t_s-T_0,[(t_A+L_s)+L_s-j-1]\}} \left[\frac{\binom{L_s-y'}{i-y'}}{\binom{L_s}{i}}\right]\frac{1}{T^2}. \quad (129)$$ where $y' = t_A + L_s - [t_B - (L_s - j)]$.

Table 5: Computation of Probabilities for 5 cases with Incentive C2F

Case	Probability
$f_1(i,j) =$	$$\sum_{t_A=t_s-T_1+L_s}^{t_s-T_0} \sum_{t_B=t_s-T_1}^{t_A-L_s} \frac{1}{T^2}. \quad (130)$$
$f_2(i,j) =$	$$\sum_{t_A=t_s-T_1}^{t_s-T_0-(2L_s-j)} \sum_{t_B=t_A+2L_s-j}^{t_s-T_0} \frac{1}{T^2}. \quad (131)$$
$f_3(i,j) =$	$$\sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_B=\max\{t_s-T_1,t_A-L_s+1\}}^{t_A} \sum_{x=\max\{0,j-(t_A-t_B)\}}^{\min\{L_s-(t_A-t_B),i,j\}} \left[\left(\frac{\binom{t_A-t_B}{j-x}\binom{L_s-(t_A-t_B)}{x}}{\binom{L_s}{j}}\right)\left(\frac{\binom{L_s-x}{i-x}}{\binom{L_s}{i}}\right)\right]\frac{1}{T^2}. \quad (132)$$
$f_4(i,j) =$	$$\sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_B=\min\{t_s-T_0,t_A+1\}}^{\min\{t_s-T_0,t_A+L_s-1\}} \sum_{x=\max\{0,j-(t_B-t_A)\}}^{\min\{L_s-(t_B-t_A),i-\min\{(t_B-t_A),(L_s-j)\},j\}} \left[\left(\frac{\binom{t_B-t_A}{j-x}\binom{L_s-(t_B-t_A)}{x}}{\binom{L_s}{j}}\right)\left(\frac{\binom{L_s-y}{i-y}}{\binom{L_s}{i}}\right)\right]\frac{1}{T^2}. \quad (133)$$ where $y = x + \min\{(t_B - t_A), (L_s - j)\}$.
$f_5(i,j) =$	$$\sum_{t_A=t_s-T_1}^{t_s-T_0-L_s} \sum_{t_B=t_A+L_s}^{\min\{t_s-T_0,[(t_A+L_s)+L_s-j-1]\}} \left[\frac{\binom{L_s-y'}{i-y'}}{\binom{L_s}{i}}\right]\frac{1}{T^2}. \quad (134)$$ where $y' = t_A + L_s - [t_B - (L_s - j)]$.

Table 6: Computation of Probabilities for 5 cases with Incentive F2C

Case	Probability
$f_1(i,j) =$	$$\sum_{t_A=t_s-T_1+L_s}^{t_s-T_0} \sum_{t_B=t_s-T_1}^{t_A-L_s} \frac{1}{T^2}. \quad (135)$$
$f_2(i,j) =$	$$\sum_{t_A=t_s-T_1}^{t_s-T_0-(2L_s-j)} \sum_{t_B=t_A+2L_s-j}^{t_s-T_0} \frac{1}{T^2}. \quad (136)$$
$f_3(i,j) =$	$$\sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_B=\max\{t_s-T_1,t_A-L_s+1\}}^{t_A} \sum_{x=\max\{0,j-(t_A-t_B)\}}^{\min\{L_s-(t_A-t_B),i,j\}} \left[\left(\frac{\binom{t_A-t_B}{j-x}\binom{L_s-(t_A-t_B)}{x}}{\binom{L_s}{j}}\right)\left(\frac{\binom{L_s-x}{i-x}}{\binom{L_s}{i}}\right)\right]\frac{1}{T^2}. \quad (137)$$
$f_4(i,j) =$	$$\sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_B=\min\{t_s-T_0,t_A+1\}}^{\min\{t_s-T_0,t_A+L_s-1\}} \sum_{x=\max\{0,j-(t_B-t_A)\}}^{\min\{L_s-(t_B-t_A),i-\min\{(t_B-t_A),(L_s-j)\},j\}} \left[\left(\frac{\binom{t_B-t_A}{j-x}\binom{L_s-(t_B-t_A)}{x}}{\binom{L_s}{j}}\right)\left(\frac{\binom{L_s-y}{i-y}}{\binom{L_s}{i}}\right)\right]\frac{1}{T^2}. \quad (138)$$ where $y = x + \min\{(t_B - t_A), (L_s - j)\}$.
$f_5(i,j) =$	$$\sum_{t_A=t_s-T_1}^{t_s-T_0-L_s} \sum_{t_B=t_A+L_s}^{\min\{t_s-T_0,[(t_A+L_s)+L_s-j-1]\}} \left[\frac{\binom{L_s-y'}{i-y'}}{\binom{L_s}{i}}\right]\frac{1}{T^2}. \quad (139)$$ where $y' = t_A + L_s - [t_B - (L_s - j)]$.

Table 7: Computation of Probabilities for 5 cases with Incentive F2F

Chapter 4

Effect of less motivated selfish peers on the Cooperation among the Dedicated Channels having Small number of Viewers

4.1 Introduction

In the current time, 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 [2], PPLive [3], Sopcast [4]. Thousands of users can simultaneously participate in these systems. Almost all live P2P video systems accommodate multiple channels (e.g., PPLive [3] can host over hundreds of channels). It is expected that, in the near future, live streaming systems with several hundreds of user-generated channels and dedicated channels will likely have thousands of live channels in total. A common practice in the live streaming systems is to organize the peers viewing the same live channel into a swarm where they form a mesh-based structure and distribute/re-distribute the streaming pieces (commonly known as chunks) to each other. Swarm formations work well when the participant peers responds generously to the upload requests of its neighbors. We propose that presence of a simple incentive mechanism will improve the continuous playback quality of the peers. First, we develop a discrete-time stochastic model for the peers of live P2P video streaming. We compare network parameters like probability of continuous playback, average download rate etc. without any selfish and with certain amount of Free Riders present in the system. Next, we introduce a simple incentive mechanism and modify our stochastic model in order to accommodate the incentive mechanism. Then we compare the result of probability of continuous playback with and without having an incentive mechanism. Our work shows that presence of an incentive mechanism improve the over all system performance.

4.2 Related Work

In the literature, there are few theoretical studies on the modeling and performance analysis of P2P systems. We first review the studies dealing with single-channel streaming systems. In one of the earliest work, Qiu *et al.* [6] developed a simple deterministic fluid model for mesh-based P2P file sharing systems which provides insights into the performance of a BitTorrent like network. The fluid model is described by a set of differential equations. Zhou *et al.* [7, 8] developed a simple probability model for data-driven systems to compare different chunk selection, downloading and peer selection strategies based on startup delay and continuity. They assumed independent (i.e., they do not know each other) and homogeneous peers in a symmetric network setting for their analytical model. Kumar *et al.* [9] provided a stochastic fluid model to explore the fundamental characteristics and limitations of swarm-based P2P streaming channels. They explored the effect of peer churn on a swarm-based channel where a large number of viewers are assumed to be either with or without a playback buffer. Massoulie *et al.* [10] studied the problem of efficient decentralized broadcasting in a P2P network with an homogeneous upload capacity and heterogeneous upload capacities, and proposed completely decentralized algorithms. Liu *et al.* [11] derived performance bounds for minimum server load, maximum streaming rate, and minimum tree depth under different peer selection constraints. Wu *et al.* [12] considered the problem of server capacity provisioning among multiple channels in order to maintain adequate streaming quality for each channel.

Jin *et al.* [13] considered a game theoric approach to show the performance improvement when the peers are motivated to cooperate more. However, they have only considered single channel in their model. Kotevski *et al.* [14] developed a hybrid model for the performance analysis of P2P live video streaming systems. They evaluated performance of the system both with and without video buffer. They used queuing network and fluid model for video streaming. However, they focused mainly on the performance issues related to the single channel P2P live video streaming systems and they did not consider any inter-channel cooperation scenario. Zhang *et al.* [15] have developed an analytical model to classify and evaluate neighbors of a peer and chunk selection strategies for the push and pull based P2P streaming systems. Based on their findings, they propose a greedy selection mechanism where a peer will select a chunk nearest to its playback pointer. Chen *et al.* [16] have developed a fluid model for analyzing the performance of P2P live video streaming systems under flash crowd. However, they mainly focused on the quality of service and issues related to peer latency and system recovery time for single channel P2P live video streaming system under flash crowd. Qiu *et al.* [17] have developed a stochastic model for performance analysis of network coding based P2P live video streaming systems. They have provided results for different segment selection techniques and compared their performances. Wu *et al.* [18] have proposed a framework and overlay design for multi-channel live video streaming which completely decouples the event of viewing a channel by a peer from the task of uploading the streaming content by the same peer. They have developed infinite-server queuing models for their system where every peer joining the system is requested to distribute streaming chunks to one or more streaming channels, where a peer is watching one channel while uploading to other channels. This puts a lot of overhead on the system to maintain many distribution groups together with their membership. Liang *et al.* [5] have proposed a partial decoupling strategy instead of a complete decoupling of the peer viewing and uploading operations in the context of multiple channels (not necessarily DCSV channels). Only some of the bandwidth-rich peers (and not all the peers) among all the channels will be assigned to a helping group and this group will be responsible for providing additional upload capacity to the peers lacking it. Unlike [18], they put lighter burden (e.g., maintaining single distribution group instead of many) on the system. However, the heavier burden is shifted on the helping peers who share their upload capacities with multiple peers from multiple channels. Liu *et al.* [19] developed a capacity model with and without node degree bound for a tree-based network. Therein, they worked on the effect of flash crowd on the requested network capacity in order to provide adequate services. Liu *et al.* [20] emphasized on content availability in the playback buffer of a peer in order to maximize the utilization of their upload capacity. In their model, they focused on minimum delay and proposed a snowball streaming algorithm for real time video streaming systems for tree-based models.

Free riders and selfish peers have been studied for long from different perspectives in different contexts. Free rider is first noted to be discussed by E.Ader *et al.* [23] in P2P context, in which the authors pointed out that there are many free riders in the network who try to take advantage of the system. Several works have been published on free riders problem in P2P File sharing systems where as there are few works done in the context of Live streaming system. Next, we present some of the works found in literature. Free riders have been studied for File sharing systems, e.g., Bittorent network by Qiu *et al.* [24], and for single channel Live streaming systems as well, e.g., in Liu *et al.* [25]. Qiu *et al.* have shown that even if the tit-for-tat mechanism is very good to protect a network from free riders in a File sharing system, the Bittorent protocol still has one option called 'unchoking' which is exploited by the free riders for their benefit. Qiu *et al.* have suggested a modification in the Bittorent protocol, more specifically in the unchocking technique and have shown this modified unchoking technique can prevent free riders. Zhengye *et al.* [25] have suggested multiple layered coding as an incentive mechanism to avoid the tendency of free riding. In this mechanism, the viewing quality of the streaming video received by a user is be determined by the contribution of upload bandwidth of that user. Koo *et al.* [27] have suggested an incentive mechanism for a P2P Content distribution system, where the jobs may be divided into smaller units, and users have incentives to truthfully revealing willingness-to-pay for services. In this mechanism, the capacity that every user makes available to others in turn determines the amount of resources that user receives. However, this mechanism is designed for the P2P File sharing schemes. Habib *et al.* [28] have developed a score-based incentive mechanism for single channel media streaming. A peer which gets higher score is rewarded with more choice of neighbor selection and flexibilities. However, it does not address the issue of fulfilling their incentive mechanism honestly. Kumar *et al.* [29] have designed a pricing and allocation mechanism for a special case of P2P network that allows the users within a firm to effectively share their resources and avoid free riding. Here, the optimal price of a task is determined based on the delay at each peer. Wu *et al.* [30] have developed an auction and bid based incentive mechanism for P2P File sharing systems in the context of social network using game theory. They also have suggested an extension of their model for real time Live streaming systems. Altman *et al.* [31] have developed a stochastic model to study behavior of the P2P networks those distribute non-authorized music, books, or articles. They even consider the presence of free riders in that network. Then this model is used to predict the efficiency of the counter-measures taken by the content providers against those P2P networks. Shahriar *et al.* [21] have developed a generic stochastic model for peers cooperating with each other where these peers are from two different channels having small number of viewers. In [22] the authors have presented another stochastic model which depicts interactions among the peers from two different channels with greater details and more realistic assumptions. In [33] the authors have presented a new stochastic model which considers the presence of free riders in live streaming systems. In this work, the authors show the impact of the free riders on the playback performance of the peers of the system. However, their work does not discuss about the impact of incentive mechanism on the free riders. Yeung *et al.* [34] have designed a tit-for-tat based incentive mechanism for single channel P2P media streaming and modeled this as two repeated game. In one of the repeated game, they have shown interactions between the streaming server and the immediate peers. The other one deals with the interaction between a peer and its neighbor. Although they have formulated an optimization model for the first case, they provided only the simulation results.

In our proposed stochastic model, we have introduced cooperation among the peers from different DCSV channels where every peer is very likely to suffer from poor channel performance. We believe that this sufferings induce a sufficient motivation for all the peers to cooperate, even if it means downloading chunks from a channel, they are not interested in viewing. Especially, since this cooperation ultimately improves the performance of their own channel, this becomes a strong incentive for the peers across DCSV channels to help each other.

Therefore, the aim of our study is to go one step further in peer collaboration, and to evaluate the improvement of the performance resulting from such a uniform cooperation. It corresponds to a realistic approach of cross-channel resource sharing and cooperation among peers from different DCSV channels.

Some of the previous works have proposed cooperation schemes in different ways at different levels. They have proposed some models but mostly provided experimental results based on heuristics. Beside the proposed HnH cooperation scheme, the core contribution of this paper is the discrete-time stochastic model in order to evaluate the performance based on collaboration among peers of DCSV channels of a HnH system. We next derive nearly closed-form expressions for performance metrics such as the probability of continuity in streaming, consumption of effective bandwidth and the average number of downloaded streaming chunks per time slot. After the validation of the model, we use this model to investigate the performance of the HnH scheme, as well as the influence of some of the network parameters: length of playback buffers, maximum delay from the source, effective upload capacity, on the performance.

Note that, some previous works have focused on bandwidth bottleneck and solved the performance problem by adding additional upload capacity to those DCSV channels. However, additional capacity can be avoided if the performance issue is interpreted as a content bottleneck problem. The solution is then to define some mechanisms in order to increase the number of participants, which will eventually increase the number of instances of the DCSV chunks. In order to reduce the additional overhead cost, we consider cooperation at the sub-stream level.

4.3 HnH Scheme

We proposed HnH, a multi-channel live streaming system which allows cooperation among performance-suffering peers of different DCSV channels. We consider that small number of participants is the key reason for poor viewing performance of the DCSV channels. More specifically, it is the absence of sufficient number of streaming chunks (rather than insufficient amount of upload capacity) that causes the problem. As a solution scheme, peers from different DCSV channels, who suffer from poor performance, can form combined channels in a HnH scheme based on multi-channels, because of their interest and mutual benefit. Once organized, all the peers viewing individual DCSV channels are considered as common members to all participating DCSV channels and each peer acquires a sufficient number of neighbors. Cooperating peers receive and distribute streaming contents of the participating channels in addition to its own viewing channel. Due to possible upload overhead on the cooperating peers in the combined channels, we propose cooperation at a sub-stream level. With the option of sub-streaming, chunks of a channel stream can be further divided into m number of sub-streams (i.e., subgroups) at the source server. For example, if the number of sub-streams is set to 5 (i.e., $m = 5$), all streaming chunks will be divided into 5 sub-streams at the server. In that case, sub-stream 1 corresponds to chunks $1, 6, 11, 16$ etc and sub-stream 2 corresponds to chunks $2, 7, 12, 17$ etc. Now, a viewing peer will always upload and download all the chunks of the stream (i.e all the sub-streams) among themselves. However, a cooperating peer may try to reduce its overhead of helping a viewing peer by only uploading and downloading the chunks corresponding to a particular sub-stream. Its cooperating buffer will contain fewer chunks than a viewing peer. In other words, instead of dealing with all the chunks of all the sub-streams, a cooperating peer may store and distribute only one of the m-sub streams of the viewing channel. In the HnH approach, a peer of a particular DCSV channel have two types of neighbors: *(i)* Viewing: peers who are watching the same channel (i.e., viewing channel) of that peer and *(ii)* Cooperating: peers from a different DCSV channel (i.e., a cooperating channel) who cooperate with a viewing peer in order to distribute its streaming chunks.

4.4 Description of the Stochastic Model

4.4.1 The Model where Selfish Peers are Unidentified

We describe the model below.

In this scheme, a peer may have two types of neighbors: *(i)* Contributors: peers who are regular peers and contributes chunks when requested and *(ii)* Free-Riders: peers who are selfish and do not contributes chunks when requested.

As illustrated in Figure 17, we identify the following interaction scenarios for a peer in this model:

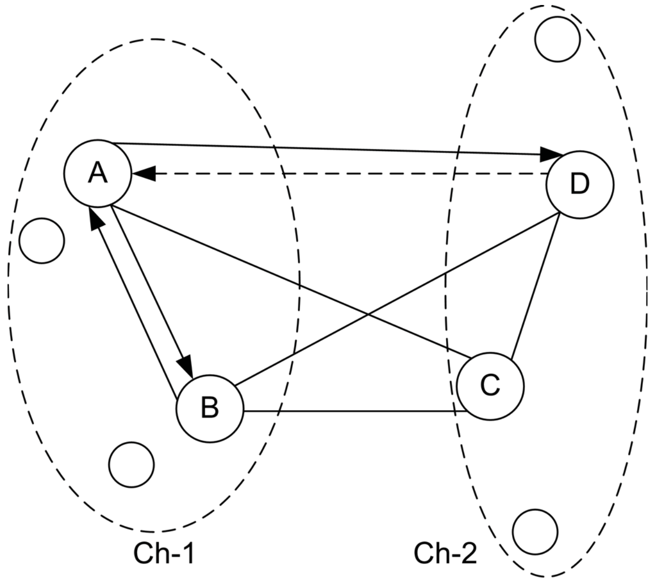


Figure 30: Peer interactions in the HnH scheme: an arrow from A to D means peer A requests and downloads streaming chunks from peer D

- **Contributing-to-Neighbors (C2N) Scenario:** In this scenario, a contributing peer responds to the request of its neighbors (including both contributors and free riders) by uploading chunks to them.
- **Free riding-to-Neighbors (F2N) Scenario:** In this scenario, a free riding peer responds to the request of its neighbors (including both contributors and free riders) by uploading chunks to them.

4.4.2 Outline of the Stochastic Model

Now, we discuss the Stochastic Model in details. We assume the peers to be homogeneous, i.e., to have identical parameters. The following notations are used in the stochastic model:
L_s : length of the viewing/playback buffer;
t_s : playback pointer (i.e., the sequence number of the chunk) at the source;
L_c : length of the cooperating buffer, i.e., buffer of the cooperating peer for its non viewing channel
$(t_s - T_0)$: the earliest playback pointer at any peer;
T : the maximum amount of delay between the playback pointer of the source and that of a peer;
$(t_s - T_1)$: the latest playback pointer at any peer;
u : the upload bandwidth of a peer;

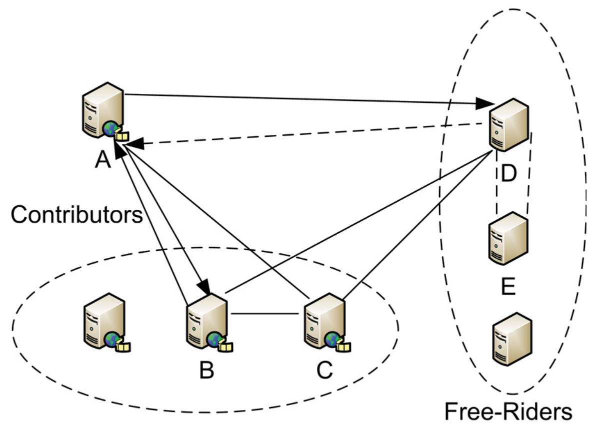


Figure 31: A Peer having contributors and Free-Rider neighbors

H_v : number of viewing neighbors of a peer
$H_c c$: number of cooperating-contributors (neighbors) of a peer
$H_c f$: total number of free riding-contributors (neighbors) of a peer;
H_f : total number of free riding neighbors of a peer;
$H = (H_c + H_f)$: total number of neighbors of a peer;
P_f : probability that a peer is a free rider;
$(1 - P_f)$: probability that a peer is a contributor (i.e., not a free rider);
q_f : probability with which a free rider acts generously as a contributor;

The model relies on the following assumptions:
- Each peer maintains a fixed number of neighbors, H. The download capacity of any peer is unlimited but the upload capacity is limited, u chunk/time slot (same for all peers).
- Each peer maintains a limited and fixed sized playback buffer, L_s, in its memory.
- A peer is said to have i useful chunks (for itself) in its playback buffer L_s, if none of these i chunks have been played yet by this particular peer. These chunks cannot be overwritten by newer chunks before they are played.
- Peers playback rate is 1 chunk/time slot and its playback pointer denotes the sequence number of the chunk the peer is playing at the current time slot. Due to limited size of the buffer, these $(L_s - i)$ old (i.e., already played) chunks can be overwritten by newer chunks.
- If a peer contains j useful chunks in its buffer L_s then $(L_s - j) \geq 0$ old chunks may also be available in the buffer at the same time.
- If a peer with i useful chunks is looking for $(L_s - i)$ chunks, all the missing $(L_s - i)$ chunks have equal priority to be fetched. More specifically, no priority is given to chunks close to

the playback pointer. Studying the effect of such a priority is part of one of our future works.
- Upload bandwidth is set to u chunks/time slot, where $u \geq 1$ is a constant. However, download bandwidth is assumed to be unlimited since in most P2P systems, normally, download bandwidth is not a bottleneck issue.
The stochastic model (as illustrated in Figure 2) is described by $(L_s + 1)$ states (of a peer), where each state corresponds to its number of useful chunks in the playback buffer. Associated with each state i (except the initial one), there is exactly one reverse transition (M_i) to the its previous state (i.e., $i - 1$) which corresponds to the playback of one chunk/time slot. From each state i, there originates $(L_s - i)$ forward transitions, $B_{i,k}$, where $(L_s - i) \geq k \geq 0$. In the subsequent sections, we discuss transition probabilities in details.

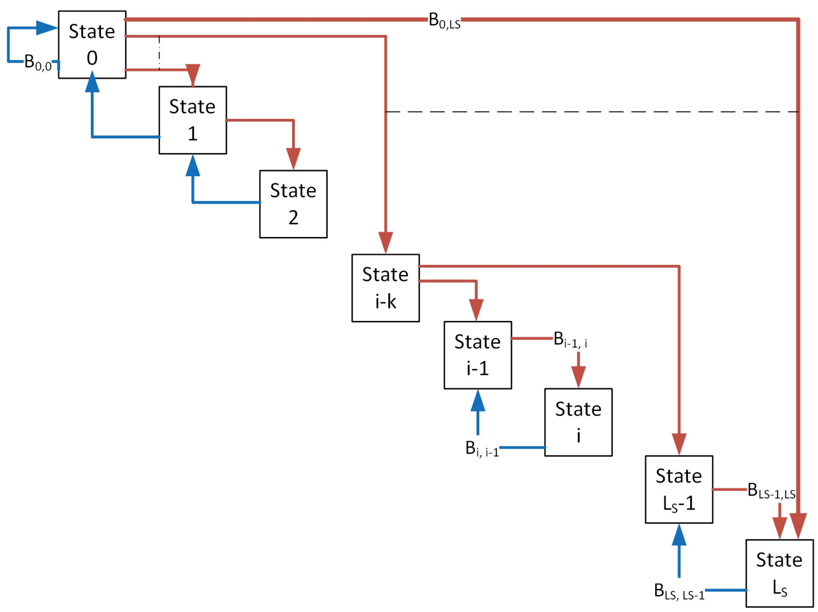


Figure 32: A discrete-time stochastic model for peers (i.e., both contributing and free riding types) having buffer size L_s.

4.4.3 Objective and Constraints:

The objective of the stochastic model is to investigate the fundamental characteristics, limitations and performance problems of a P2P live streaming system having free riders, depending on the values of the network parameters (e.g., buffer size, maximum delay, number of free riders). After computing the transition probabilities, we will be able to estimate the probability distribution of the number of useful chunks. From these probability distributions, we calculate the probability of continuous playback and gain more information on how the continuous playback performance of such a HnH based system is controlled by the network parameters (e.g., buffer size, maximum delay, number of free riders). We have the following constraints:

- A streaming chunk, k, will be stored by peer A if it satisfies: $t_A \leq k \leq (t_A + L_s - 1)$ where, t_A is the playback pointer of A, otherwise, it will be discarded.
- A peer having i useful chunks in its buffer L_s sends a maximum of $\min\{H, (L_s - i)\}$ requests to its H neighbors in the current time slot.
- The playback pointer t_A of peer A lies within the following limit: $(t_s - T_1) \leq t_A \leq (t_s - T_0)$.

4.5 Analysis of the Stochastic Model

Now, we analyze the stochastic model and calculate the probabilities for scenarios mentioned in Section 4.4. In a live streaming channel, the request of a peer to a neighbor in order to get a chunk, depends on the relative positions of their playback pointers and the playback buffer. We have the following two categories of positions:

(1) No overlapping of chunks from the corresponding buffers of peer A with its neighbor B and (2) Some overlapping of chunks from the corresponding buffers of A and B. Category 1 can be further sub-divided into two sub-categories:

(1.1) Peer A is playing before neighbor B and there is no overlapping of chunks in their buffers (i.e., peer A has already played all the chunks of its neighbor B) and (1.2) Neighbor B is playing much ahead of peer A such that even the old chunks of B are too new for peer A to request (i.e., there is no overlapping). Category 2 can be further sub-divided into three sub-categories:

(2.1) Peer A is playing ahead of its neighbor B, having some overlapping chunks between A and B, (2.2) Neighbor B is playing ahead of peer A and there are some overlapping chunks between A and B and (2.3) Neighbor B is playing much ahead of peer A however, there is some overlapping of old chunks of B with the new chunks of A. These five general categories are illustrated in Figure 33 and Figure 34. We next derive probabilities of the stochastic model for the C2C scenario.

4.6 Analysis of the Stochastic Model without any Incentive

We assume that currently there is no way to know which peer is a contributing peer and which peer is a free riding peer. Consider an arbitrary peer (e.g., peer A) and one of its

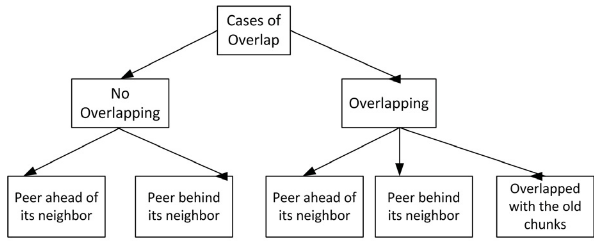


Figure 33: Cases of relative positions of peer A and its neighbor B where t_A and t_B.

neighbor (e.g., peer B) such that they may be any combination of contributing and free riding peers. Suppose, at the current time slot, peer A has i and its neighbor B has j useful chunks in their respective buffers. Based on that, we will calculate, U_i, probability that peer A having i useful chunks in its buffer in a specific time slot, will be interested to get a chunk from a random neighbor, B.

$$U_i^{c2c} = \sum_{j=0}^{L_s} \left(1 - f^{c2c}(i,j)\right) P_j^{c2c}. \tag{140}$$

where P_j^{c2c} is the probability that a random contributing peer in the network has j useful chunks. In order to calculate, U_i, first we define $f(i,j)$, to be the probability that peer A with i useful chunks in its buffer is not interested in any of the j useful chunks of B. We calculate $f(i,j)$ for the five mutually exclusive cases:

$\text{Case}_1 : t_s - T \leq t_B \leq t_A - L_s$

$\text{Case}_2 : t_A + 2L_s - j \leq t_B \leq t_s - 1$

$\text{Case}_3 : t_A - L_s + 1 \leq t_B \leq t_A$

$\text{Case}_4 : t_A + 1 \leq t_B \leq t_A + L_s - 1$

$\text{Case}_5 : t_A + L_s \leq t_B \leq t_A + 2L_s - j - 1.$

We, now calculate the probability for each case, such that peer A will not be interested in the chunks of its neighbor B.

Let $f_k(i,j)$ be the probability in Case_k, where $k = 1...5$, such that, in the current time slot, peer A is not interested in the chunks of its neighbor B under the condition of $Case_k$. We assume that the playback pointer of a peer is uniformly distributed within the interval $[t_s - T_0, t_s - T_1]$ and $T = (T_1 - T_0 + 1)$. So, the probability that the playback pointer of peer

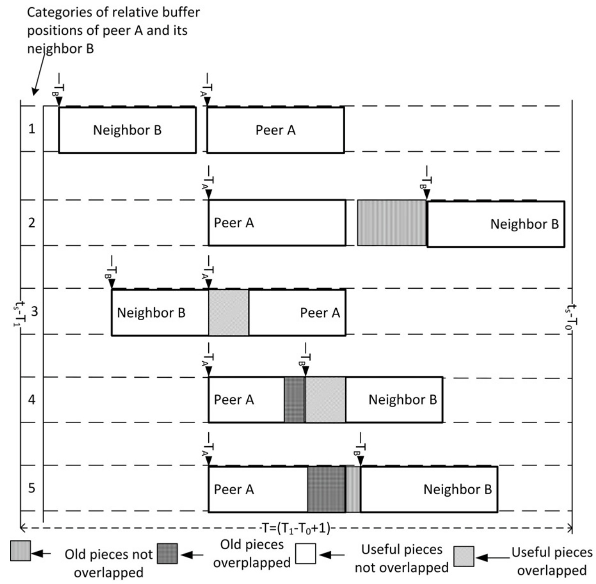


Figure 34: Relative positions of peer A and its neighbor B where t_A and t_B denote the playback pointer of the peer A and its neighbor B, respectively.

A is in a given position is $1/T$. It is to be noted that, the definition of the probability of the playback pointer of any peer to be in a given position will be same for all the cases and the model will work with other distributions as well, however it will be more complicated with those distributions.

As the $f_k(i,j)$ probabilities are mutually exclusive:

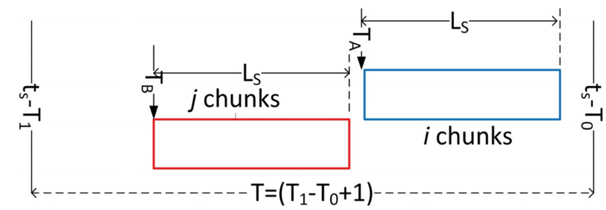
$$f(i,j) = f_1(i,j) + f_2(i,j) + f_3(i,j) + f_4(i,j) + f_5(i,j). \quad (141)$$

Computations of the $f_k(i,j)$ probabilities are as follows.

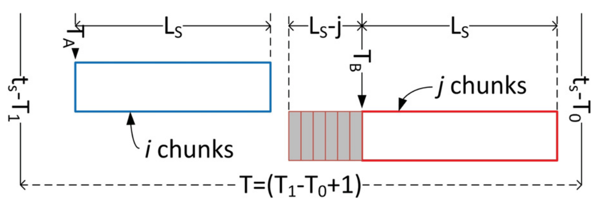
$Case_1^{c2c} : t_s - T_1 \leq t_B \leq t_A - L_s$ (see Figure 35(a)):

Peer A has i useful chunks and its random neighbor B has j useful chunks in their respective buffers. Moreover, those buffers have no content overlap. Peer A has already played all the j chunks of B and has no further interest in them. We suppose $\overline{f}_1^{c2c}(i,j)$, be the probability that A is not interested in B (i.e., irrespective of it the interval, T). Under the same assumption, the definition of the probability of the playback pointer of any peer to be in a given position will be same for all the subsequent cases. It follows:

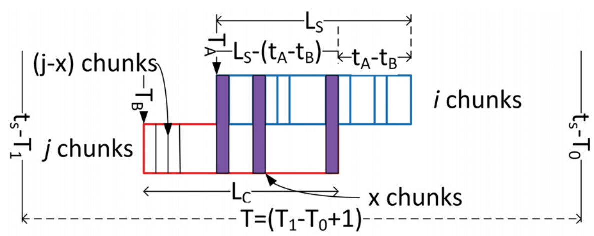
$$f_1^{c2c}(i,j) = \sum_{t_A=t_s-T_1+L_s}^{t_s-T_0} \sum_{t_B=t_s-T_1}^{t_A-L_s} \frac{1}{T^2}. \quad (142)$$



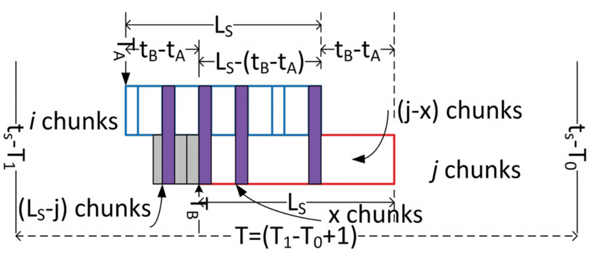
(a) $Case_1^{c2c}$



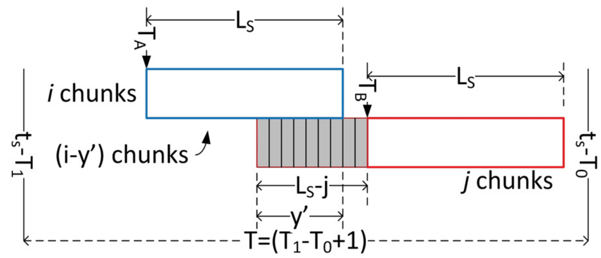
(b) $Case_2^{c2c}$



(c) $Case_3^{c2c}$



(d) $Case_4^{c2c}$



(e) $Case_5^{c2c}$

Figure 35: Overlapping of i useful streaming chunks of peer A with j useful chunks of one its random neighbor B in the contributing-to-contributing (c2c) cases

In this case, probability that A is not interested in the chunks of $B = 1$. Here, probability that the playback pointer, t_A, of peer A is at a given position is $1/T$ and the probability that the playback pointer t_B, of peer B is at a given position is also $1/T$.

$Case_2^{c2c} : t_A + 2L_s - j \leq t_B \leq t_s - T_0$ (see Figure 35(b)):

In this case, buffers of peer A and neighbor B have no content overlap and as opposed to $Case_1^{c2c}$, buffer of the neighbor B contains more recent pieces than peer A. The lowest sequence number of the $(L_s - j)$ old chunks (residing in the buffer) of peer B is even higher than the highest sequence number of the chunks that A can store in its buffer in the current time slot. Hence, peer A has no interest in its neighbor B. It follows:

$$f_2^{c2c}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0-(2L_s-j)} \sum_{t_B=t_A+2L_s-j}^{t_s-T_0} \frac{1}{T^2}. \tag{143}$$

$Case_3[t_A - L_s + 1 \leq t_B \leq t_A]$ (see Figure 35(c)):

Among the j useful chunks of neighbor B, x chunks are already present in the overlapping portion of the buffer of peer A. If A is not interested in B, then whatever chunks neighbor B has in the overlapping portion, peer A already has those chunks. The remaining $(j - x)$ chunks of B are of no interest to A (as those chunks are already played by A). It follows:

$$f_3(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_B=\max\{t_s-T_1, t_A-L_s+1\}}^{t_A} \sum_{x=\max\{0, j-(t_A-t_B)\}}^{\min\{L_s-(t_A-t_B), i, j\}} \left[\left(\frac{\binom{t_A-t_B}{j-x}\binom{L_s-(t_A-t_B)}{x}}{\binom{L_s}{j}} \right) \left(\frac{\binom{L_s-x}{i-x}}{\binom{L_s}{i}} \right) \right] \frac{1}{T^2}. \tag{144}$$

In this case, probability that A is not interested in the chunks of B is computed inside the square braces. The first part inside the square braces denotes the probability that neighbor B has j chunks in its buffer and among these j chunks, x chunks are in the portion (i.e., $L_s - (t_A - t_B)$) which is overlapped with the buffer of peer A. Remaining $j - x$ chunks of peer B lies in the non-overlapped region, $t_A - t_B$. The second part denotes that peer A has i chunks where these i chunks include all the x chunks which are in the buffer of neighbor B.

$Case_4 : t_A + 1 \leq t_B \leq t_A + L_s - 1$ (see Figure 35(d)):

Neighbor B has x (out of j) useful chunks in the overlapping portion. If peer A is not interested in B then A also has all the x chunks in its buffer. Moreover, peer A has all the $(L_s - j)$ old chunks of B. The remaining $(j - x)$ chunks of B are of no interest to A (as those chunks are too new for it for the current time slot). We then have:

$$f_4(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_B=\min\{t_s-T_0,t_A+1\}}^{\min\{t_s-T_0,t_A+L_s-1\}} \sum_{x=\max\{0,j-(t_B-t_A)\}}^{\min\{L_s-(t_B-t_A),i-\min\{(t_B-t_A),(L_s-j)\},j\}} \left[\left(\frac{\binom{t_B-t_A}{j-x}\binom{L_s-(t_B-t_A)}{x}}{\binom{L_c}{j}}\right)\left(\frac{\binom{L_s-y}{i-y}}{\binom{L_s}{i}}\right)\right]\frac{1}{T^2}. \quad (145)$$

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

Here, probability that A is not interested in the chunks of B is computed inside the square braces where the first part inside the square braces denotes the probability that neighbor B has j chunks in its buffer and among these j chunks, x chunks are in the portion, $L_s-(t_B-t_A)$, which is overlapped with the buffer of peer A. Remaining $j-x$ useful chunks lies in the non-overlapped region, $t_B - t_A$, of peer B. However, in this case, B has $L_s - j$ old chunks which are overlapped with the first $t_B - t_A$ portion of the buffer of peer A The second part denotes that peer A has i chunks where these i chunks include all the x useful chunks and all the $L_s - j$ old chunks of the buffer of neighbor B (i.e., defined under y).

$Case_5^{c2c}$: $t_A + L_s \leq t_B \leq t_A + 2L_s - j - 1]$(see Figure 35(e)):

In this case, there is a overlapping between the i useful chunks of peer A and y' (among $(L_s - j)$) old chunks of neighbor B. Peer A is not interested in the j useful chunks of neighbor B (as those are too new for A in the current time slot). We have:

$$f_5^{c2c}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0-L_s} \sum_{t_B=t_A+L_s}^{\min\{t_s-T_0,[(t_A+L_s)+L_s-j-1]\}} \left[\frac{\binom{L_s-y'}{i-y'}}{\binom{L_s}{i}}\right]\frac{1}{T^2}. \quad (146)$$

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

Summing the probabilities of all 5 cases leads to the following expression for $f(i,j)$:

$$f(i,j) = f_1(i,j) + f_2(i,j) + f_3(i,j) + f_4(i,j) + f_5(i,j). \quad (147)$$

We recall the definition of U_i and express as follows:

$$U_i = \sum_{j=0}^{L_s} (1 - f(i,j)) \times P_j. \quad (148)$$

where, P_j is the probability that a random peer in the network has j useful chunks. Next, we calculate the probabilities of downloading chunks.

Probabilities of requesting and downloading chunks

We recall that, any random peer in this network has H fixed number of neighbors and a buffer L_s of fixed length. It is also assumed that at any time slot if any randomly picked peer (e.g. peer A) receives more than one request, u of them (including the one from A) will be fulfilled randomly by peer B. We define $s_{\max}$, the maximum number of requests that an arbitrary peer (e.g., A) having i chunks in the current time slot can send to its neighbors as: $s_{\max} = \min\{H, (L_s - i)\}$ where, $(L_s - i)$ is maximum number of chunks that peer D wants to download at that specific time slot and $H = (H_c + H_f)$.

Next, we define $F(H, i', k)$, as the probability that a randomly selected peer which has i useful chunks in its buffer, in a specific time slot, looking for $i' = (L_s - i)$, chunks in that time slot, sends k requests to its neighbors. $F(H, i', k)$ can be recursively calculated as follows:

$$F(H, i', k) = U_i F(h - 1, i' - 1, k - 1) + (1 - U_i) F(h - 1, i', k) \quad (149)$$

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

The first component of the right hand side in (149) assumes the peer receives one of the requested chunks from one of the H viewing neighbors. Hence, in the second part, it is looking for one less chunk (i.e., $i' - 1$) from the remaining neighbors (i.e., $h - 1$) and sends one less request (i.e., $k - 1$).

The recurrence relation (149) has the following initial conditions:

$$F(H, i', k) = \begin{cases} 0 & \text{if } (h = 0 \text{ or } i' = 0) \text{ and } k > 0 \\ 1 & \text{if } (h = 0 \text{ or } i' = 0) \text{ and } k = 0. \end{cases}$$

Next, we calculate (i) the expected-value of $F(h, i', k)$ as $\overline{L}_i$ and (ii) define $\overline{k}$ as the average number of requests that any arbitrary cooperating peer sends to its contributing neighbors at any given time slot.

$$\overline{L}_i = \sum_{k=0}^{\min\{i', H\}} k F(h, i', k); \quad (150)$$

$$\overline{k} = \sum_{i=0}^{L_s} P_i \overline{L}_i. \quad (151)$$

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

$$\overline{X} = \frac{(H-1)\overline{k}}{H} \tag{152}$$

This shows that $H-1$ neighbors send $H\overline{k}$ requests to their neighbors and on an average each of the H neighbors receive $H\overline{k}/H$ requests from them.

4.6.1 Probabilities of Fulfilling Requests by the Contributing Peers and the Selfish Peers

We assume here that a peer will have both viewing and cooperating neighbors at the same time. Next, we define the following:
(i) Q^{c2n} as the probability that a randomly selected contributing peer (e.g., peer B) arbitrarily fulfills u requests (i.e., including the one from peer A) among all the requests it received:

$$Q^{c2n} = \frac{u_v}{1+\overline{\overline{X}}} \tag{153}$$

Similarly, we define:
(ii) Q^{f2n} as the probability that a randomly selected free riding peer (e.g., peer D) arbitrarily fulfills u requests (i.e., including the one from peer A) among all the requests it received:

$$Q^{f2n} = \frac{u}{1+\overline{\overline{X}}} \times q_f \tag{154}$$

Now, we compute the probability transition in the discrete-time stochastic model shown in Figure 2 as follows:
We define, $r_{i,n}$, probability that a viewing peer having i useful chunks at a given time slot, downloads n chunks in the same time slot. Hence, depending on the type of neighbors (i.e., contributing and free riding), $r_{i,n}$ can be expressed as follows:

$$\begin{aligned} r_{i,n} = \sum_{k=n}^{\min\{L_s-i,H\}} \sum_{k_f=0}^{k} F(H, L_s-i, k) \\ \left[\binom{k}{k_f}(P^f)^{k_f}(1-P^f)^{(k-k_f)}\right] \\ \sum_{l=0}^{\min\{n,k_f\}} \left[\binom{k_f}{l}(Q^{f2n})^{l}(1-Q^{f2n})^{(k_f-l)}\right] \\ \left[\binom{k-k_f}{n-l}(Q^{c2n})^{(n-l)}(1-Q^{c2n})^{(k-k_f-n+l)}\right] \end{aligned} \tag{155}$$

Next, we define, M_i, probability that a randomly picked peer in the network with the buffer L_s and i distributed useful chunks, in its buffer in a specific time slot, can play a chunk in the same time slot. Since we do not set priority to any of the chunks, the probability that we have as follows:

$$M_i = \frac{i}{L_s} \qquad i = 0, 1, ..., L_s. \tag{156}$$

If the system is in state P_i, the probability that the system jumps to state P_{i+k} in the next time slot is given by:

$$B_{i,i+k} = (1 - M_i)r_{i,k} + M_i r_{i,k+1} \tag{157}$$

The first term on the right hand side corresponds to a situation where the first chunk is missing (i.e., the peer cannot playback) and k chunks are downloaded. The second term corresponds to a situation where the peer plays the first chunk and downloads $(k+1)$ chunks in the current time slot. When the system is in steady state, the probability distribution of any state in the model (shown in Figure 2) does not change with time and we get the following relation:

$$B_{i+1,i} \times P_{i+1} + \left(\sum_{k=0}^{\min\{H,i\}} (B_{i-k,i} \times P_{i-k}) \right) - \sum_{k=1}^{L_s - i} (B_{i,i+k} \times P_i) - B_{i,i-1} \times P_i = 0. \tag{158}$$

It is difficult to find a closed form of the solution of equation (158) in order to get the peer distribution $\{P_i\}$. However, it is possible to solve (158) numerically.

Next, we define P_{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 chunk at this time slot. It can be expressed as follows:

$$P_{cont} = \sum_{i=0}^{L_s} P_i \times M_i. \tag{159}$$

4.7 A Simple Incentive for the Peers to Cooperate

In this section we assume that there is a simple incentive mechanism for the peers (specially for the free-riders) to cooperate each other in a P2P live streaming system. In the previous model in Section 3.5 where there was no incentive mechanism and there was no monitoring of how generously the neighbors are granting the request from a peer. We propose a simplified incentive mechanism in order to use for our analysis. Later on, we can use this general analysis for more specific cases of incentive mechanisms. We develop our next stochastic model for a P2P system having such an incentive mechanism. We assume that each peer maintains a rank R_j for each of its neighbors: Rank of a viewing neighbor j:

$$R_j = \frac{n_j}{1 + N_j}. \tag{160}$$

4.8 Analysis of the Stochastic Model with Incentive

We analyze the stochastic model and calculate the probabilities for appropriate scenarios, C2C, C2F, F2C, as mentioned in Section 4.4. In this model we assume that the probability distribution the streaming chunks present in the buffer of a free-rider is different from that of a contributing peer. We calculate this probability distribution in the $F2C$ interaction.

4.8.1 Contributing-to-Contributing (C2C) Peers' Interactions

In the C2C scenario, consider an arbitrary peer (e.g., peer A) and one of its neighbors (e.g., peer B) such that both are with playback buffer of equal size, L_s. Suppose, at the current time slot, peer A has i useful chunks and neighbor B has j useful chunks in their respective buffers. t_A and t_B are the sequence numbers of the chunks to be played by the corresponding peer A and its neighbor B. Based on that, we calculate, U_i^{c2c}, the probability that peer A, while having i useful chunks in its buffer in a specific time slot, is interested in getting a chunk from a random neighbor, B. In order to calculate, U_i^{c2c}, let $f^{c2c}(i,j)$ be the probability that, at a given time slot, peer A with i useful chunks in its buffer is not interested in any of the j useful chunks of B. We then have:

$$U_i^{c2c} = \sum_{j=0}^{L_s} \left(1 - f^{c2c}(i,j)\right) P_j^{c2c}. \tag{161}$$

where P_j^{c2c} is the probability that a random contributing peer in the network has j useful chunks. We calculate $f^{c2c}(i,j)$ for each of the following five mutually exclusive cases:
$\text{Case}_1^{c2c} : t_s - T_1 \leq t_B \leq t_A - L_s$
$\text{Case}_2^{c2c} : t_A + 2L_s - j \leq t_B \leq t_s - T_0$
$\text{Case}_3^{c2c} : t_A - L_s + 1 \leq t_B \leq t_A$
$\text{Case}_4^{c2c} : t_A + 1 \leq t_B \leq t_A + L_s - 1$
$\text{Case}_5^{c2c} : t_A + L_s \leq t_B \leq t_A + 2L_s - j - 1.$

We denote by $f_k^{c2c}(i,j)$ be the probability that peer A is not interested in the useful pieces of peer B and these two peers are in Case_k^{c2c}, where $k = 1...5$. We suppose $\overline{f}_1^{c2c}(i,j)$, be the probability that A is not interested in B (i.e., irrespective of it the interval, T). We assume

that the playback pointer of a peer is uniformly distributed in interval $[t_s - T_0, t_s - T_1]$ and $T = (T_1 - T_0 + 1)$. So, the probability that the playback pointer of peer A is in a given position is $1/T$. It is to be noted that, the definition of the probability of the playback pointer of any peer to be in a given position will be same for all the cases and the model will work with other distributions as well, however it will be more complicated
It follows:

$$f_k^{c2c}(i,j) = \overline{f}_1^{c2c}(i,j) \cdot \frac{1}{T^2}. \quad (162)$$

Under these assumption, the definition of $f_k^{c2c}(i,j)$ will be same for all the cases in Case_k^{c2c}, where $k = 1...5$. As the $f_k^{c2c}(i,j)$ probabilities are mutually exclusive:

$$f^{c2c}(i,j) = f_1^{c2c}(i,j) + f_2^{c2c}(i,j) + f_3^{c2c}(i,j) + f_4^{c2c}(i,j) + f_5^{c2c}(i,j). \quad (163)$$

Computations of the $f_k^{c2c}(i,j)$ probabilities are as follows.
$Case_1^{c2c} : t_s - T_1 \leq t_B \leq t_A - L_s$ (see Figure 35(a)):
Peer A has i useful chunks and its random neighbor B has j useful chunks in their respective buffers. Moreover, those buffers have no content overlap. Peer A has already played all the j chunks of B and has no further interest in them. We suppose $\overline{f}_1^{c2c}(i,j)$, be the probability that A is not interested in B (i.e., irrespective of it the interval, T). We assume that the playback pointer of a peer is uniformly distributed in interval $[t_s - T_0, t_s - T_1]$ and $T = (T_1 - T_0 + 1)$. So, the probability that the playback pointer of peer A is in a given position is $1/T$. It is to be noted that, the model will also work with other distributions, however it will be more complicated
Under the same assumption, the definition of the probability of the playback pointer of any peer to be in a given position will be same for all the subsequent cases. It follows:

$$f_1^{c2c}(i,j) = \sum_{t_A = t_s - T_1 + L_s}^{t_s - T_0} \sum_{t_B = t_s - T_1}^{t_A - L_s} \frac{1}{T^2}. \quad (164)$$

In this case, $\overline{f}_1^{c2c}(i,j)$=1
$Case_2^{c2c} : t_A + 2L_s - j \leq t_B \leq t_s - T_0$ (see Figure 35(b)):
In this case, buffers of peer A and neighbor B have no content overlap and as opposed to $Case_1^{c2c}$, buffer of the neighbor B contains more recent pieces than peer A. The lowest sequence number of the $(L_s - j)$ old chunks (residing in the buffer) of peer B is even higher than the highest sequence number of the chunks that A can store in its buffer in the current

time slot. Hence, peer A has no interest in its neighbor B. It follows:

$$f_2^{c2c}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0-(2L_s-j)} \sum_{t_B=t_A+2L_s-j}^{t_s-T_0} \frac{1}{T^2}. \tag{165}$$

In this case, $\overline{f}_2^{c2c}(i,j)$=1

$Case_3^{c2c}[t_A - L_s + 1 \leq t_B \leq t_A]$ (see Figure 35(c)):

Among the j useful chunks of neighbor B, x chunks are already present in the overlapping portion of the buffer of peer A. If A is not interested in B, then whatever chunks neighbor B has in the overlapping portion, peer A already has those chunks. The remaining $(j-x)$ chunks of B are of no interest to A (as those chunks are already played by A). It follows:

$$f_3^{c2c}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_B=\max\{t_s-T_1, t_A-L_s+1\}}^{t_A} \sum_{x=\max\{0, j-(t_A-t_B)\}}^{\min\{L_s-(t_A-t_B), i, j\}} \left[\left(\frac{\binom{t_A-t_B}{j-x}\binom{L_s-(t_A-t_B)}{x}}{\binom{L_s}{j}}\right)\left(\frac{\binom{L_s-x}{i-x}}{\binom{L_s}{i}}\right)\right]\frac{1}{T^2}. \tag{166}$$

In this case, $\overline{f}_3^{c2c}(i,j)$ is calculated inside the square braces. The first part of $\overline{f}_3^{c2c}(i,j)$ denotes the probability that neighbor B has j chunks in its buffer and among these j chunks, x chunks are in the portion which is overlapped with the buffer of peer A. The second part denotes that peer A has i chunks where these i chunks include all the x chunks which are in the buffer of neighbor B. Under the same assumption, $\overline{f}_k^{c2c}(i,j)$ will be calculated in the same manner for all the subsequent cases.

$Case_4^{c2c}: t_A + 1 \leq t_B \leq t_A + L_s - 1$ (see Figure 4(d)):

Neighbor B has x (out of j) useful chunks in the overlapping portion, however, peer A already has all the x chunks in its buffer. Moreover, peer A has all the $(L_s - j)$ old chunks of B. The remaining $(j-x)$ chunks of B are of no interest to A (as those chunks are too new for it for the current time slot). We then have:

$$f_4^{c2c}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_B=\min\{t_s-T_0,t_A+1\}}^{\min\{t_s-T_0,t_A+L_s-1\}} \sum_{x=\max\{0,j-(t_B-t_A)\}}^{\min\{L_s-(t_B-t_A),i-\min\{(t_B-t_A),(L_s-j)\},j\}} \left[\left(\frac{\binom{t_B-t_A}{j-x}\binom{L_s-(t_B-t_A)}{x}}{\binom{L_s}{j}}\right)\left(\frac{\binom{L_s-y}{i-y}}{\binom{L_s}{i}}\right)\right]\frac{1}{T^2}. \quad (167)$$

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

$Case_5^{c2c} : t_A + L_s \leq t_B \leq t_A + 2L_s - j - 1]$(see Figure 4(e)):

In this case, there is a overlapping between the i useful chunks of peer A and y' (among $(L_s - j)$) old chunks of neighbor B. Peer A is not interested in the j useful chunks of neighbor B (as those are too new for A in the current time slot). We have:

$$f_5^{c2c}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0-L_s} \sum_{t_B=t_A+L_s}^{\min\{t_s-T_0,[(t_A+2L_s)-j-1]\}} \left[\frac{\binom{L_s-y'}{i-y'}}{\binom{L_s}{i}}\right]\frac{1}{T^2}. \quad (168)$$

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

4.8.2 Contributing-to-Selfish (C2S) Peers' Interactions

In the c2f scenario, consider an arbitrary peer (e.g., peer A with playback buffer L_s) and one of its free-riding neighbors (e.g., peer D with playback buffer L_s). Suppose, at the current time slot, peer A has i useful chunks and neighbor D has j useful chunks in their respective buffers. Based on that, we will calculate, U_i^{c2f}, probability that peer A having i useful chunks in its buffer in a specific time slot, is interested to get a chunk from a random cooperative neighbor D. In order to calculate, U_i^{c2f}, let $f^{c2f}(i,j)$ be the probability that, at a given time slot, peer A with i useful chunks in its buffer is not interested in any of the j useful chunks of D. We then have:

$$U_i^{c2f} = \sum_{j=0}^{L_s} \left(1 - f^{c2f}(i,j)\right) P_j^{c2f}. \quad (169)$$

where P_j^{c2f} is the probability that a random peer in the network has j useful chunks.

We denote by $f_k^{c2f}(i,j)$ the $f^{c2f}(i,j)$ probability in Case$_k^{c2f}$, where $k = 1...5$. As the

$f_k^{c2f}(i,j)$ probabilities are mutually exclusive:

$$f^{c2f}(i,j) \quad = \quad f_1^{c2f}(i,j) \ + \ f_2^{c2f}(i,j) \ + \ f_3^{c2f}(i,j) \ + \ f_4^{c2f}(i,j) \ + \ f_5^{c2f}(i,j). \quad (170)$$

Computations of the $f_k^{c2f}(i,j)$ probabilities are as follows.
Case$_1^{c2f}$: $t_s - T_1 \leq t_D \leq t_A - L_s$
Case$_2^{c2f}$: $t_A + 2L_s - j \leq t_D \leq t_s - T_0$
Case$_3^{c2f}$: $t_A - L_s + 1 \leq t_D \leq t_A$
Case$_4^{c2f}$: $t_A + 1 \leq t_D \leq t_A + L_s - 1$
Case$_5^{c2f}$: $t_A + L_s \leq t_D \leq t_A + 2L_s - j - 1$.
We, now calculate the probability, that peer A will not be interested in the chunks of neighbor D.
Let $f_k^{c2f}(i,j)$ be the probability in Case$_k^{c2f}$, where $k = 1...5$, such that, in the current time slot, peer A is not interested in the streaming chunks of neighbor D under the condition of $Case_k^{c2c}$.
$Case_1^{c2f}$ $[t_s - T_1 \leq t_D \leq t_A - L_s]$ (see Figure 5(a)):
Peer A has i useful chunks in its buffer, L_s and free-riding neighbor D has j useful chunks in its buffer, L_s. The relative position of their playback pointers are such that there is no overlapping of contents between the buffers of these two peers. Peer A has already played back all the j chunks residing in the buffer L_s of neighbor D and has no further interest in them.

$$f_1^{c2f}(i,j) = \sum_{t_A=t_s-T_1+L_s}^{t_s-T_0} \sum_{t_D=t_s-T_1}^{t_A-L_s} \frac{1}{T^2} \quad (171)$$

Case$_2^{c2f}$: $t_A + 2L_s - j \leq t_D \leq t_s - T_0$ (see Figure 5(b)):
In this case, there is no overlapping of the two buffers of peer A and the free-riding neighbor D. As opposed to $Case_1^{c2f}$, buffer of the neighbor D contains more recent pieces than peer A. The lowest sequence number of the $(L_s - j)$ old chunks of peer D is even higher than the highest sequence number of the chunks that A can store in its buffer in the current time slot.

$$f_2^{c2f}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0-(2L_s-j)} \sum_{t_D=t_A+2L_s-j}^{t_s-T_0} \frac{1}{T^2} \quad (172)$$

Case$_3^{c2f}$: $t_A - L_s + 1 \leq t_D \leq t_A$ (see Figure 5(c)):
contributing peer A has i useful pieces and free-riding neighbor D has j useful pieces in their corresponding buffers. Among the j chunks of free-riding neighbor D, x are already present in the overlapping portion of the buffer (with peer A). More specifically, whatever chunks

(i.e., x) peer D has in the overlapping portion, peer A also has those chunks in its buffer. The remaining $(j-x)$ chunks are of no interest to A (as playback pointer of A has already passed those chunks).

$$f_3^{c2f}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_D=\max\{t_s-T_1,t_A-L_c+1\}}^{t_A} \sum_{x=\max\{0,i-(t_A-t_D)\}}^{\min\{L_s-(t_A-t_D),i,j\}} \left[\left(\frac{\binom{(t_A-t_D)}{j-x}\binom{(L_s-(t_A-t_D))}{x}}{\binom{L_s}{j}} \right) \left(\frac{\binom{L_s-x}{i-x}}{\binom{L_s}{i}} \right) \right] \frac{1}{T^2} \quad (173)$$

Case$_4^{c2f}$: $t_A+1 \le t_D \le t_A+L_s-1$ (see Figure 5(d)):
free-riding peer D has j useful chunks and (L_s-j) old chunks in its buffer, L_s. Among the j useful chunks, D has x chunks in the overlapping buffer and A has all these x. Moreover, peer A has all the (L_s-j) old chunks of D. The remaining $(j-x)$ chunks are of no interest to A (as those chunks are too new for A for the current time slot). $Case_4^{c2c}$: $t_A+1 \le t_B \le t_A+L_s-1$ (see Figure 4(d)):
Neighbor B has x (out of j) useful chunks in the overlapping portion, however, peer A already has all the x chunks in its buffer. Moreover, peer A has all the (L_s-j) old chunks of B. The remaining $(j-x)$ chunks of B are of no interest to A (as those chunks are too new for it for the current time slot). We then have:

$$f_4^{c2f}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_D=\min\{t_s-T_0,t_A+1\}}^{\min\{t_s-T_0,t_A+L_s-1\}} \sum_{x=\max\{0,j-(t_D-t_A)\}}^{\min\{L_s-(t_D-t_A),i-\min\{(t_D-t_A),(L_s-j)\},j\}} \left[\left(\frac{\binom{t_D-t_A}{j-x}\binom{L_s-(t_D-t_A)}{x}}{\binom{L_s}{j}} \right) \left(\frac{\binom{L_s-y}{i-y}}{\binom{L_s}{i}} \right) \right] \frac{1}{T^2}. \quad (174)$$

where $y = x + \min\{(t_D-t_A),(L_s-j)\}$.
Case$_5^{c2f}$: $t_A+L_s \le t_D \le t_A+2L_s-j-1$ (see Figure 5(e)):
contributing peer A has i useful chunks in its buffer and the free-riding neighbor D has j useful chunks and (L_s-j) old chunks in its buffer. In this case, there is no overlapping among the useful chunks of peer A and its neighbor D. However, the i useful chunks of peer

A has overlapping with y' (out of $(L_s - j)$) old chunks of D and A has all the y' chunks in its buffer. Peer A will not be interested about the j chunks of neighbor D (as those chunks are too new for the current time slot).

$$f_5^{c2f}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0-L_s} \sum_{t_D=t_A+L_s}^{(t_A+2L_s)-j-1} \left[\frac{\binom{L_s-y'}{i-y'}}{\binom{L_s}{i}} \right] \frac{1}{T^2}. \quad (175)$$

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

4.8.3 Selfish-to-Contributing (S2C) Peers' Interactions

In the f2c scenario, consider an arbitrary free-riding peer (e.g., peer D with playback buffer L_s) and one of its contributing neighbors (e.g., peer A with playback buffer L_S) such that they are viewers of two different channels. Suppose, at the current time slot, peer D has i useful chunks and its neighbor A has j useful chunks in their respective buffers. Based on that, we will calculate, U_i^{f2c}, probability that peer D having i useful chunks in its buffer in a specific time slot, is interested to get a chunk from a random contributing neighbor A. In order to calculate, U_i^{f2c}, let $f^{f2c}(i,j)$ be the probability that, at a given time slot, peer D with i useful chunks in its buffer is not interested in any of the j useful chunks of contributing neighbor A. We then have:

$$U_i^{f2c} = \sum_{j=0}^{L_s} \left(1 - f^{f2c}(i,j)\right) P_j^{f2c}. \quad (176)$$

where P_j^{f2c} is the probability that a random peer in the network has j useful chunks.

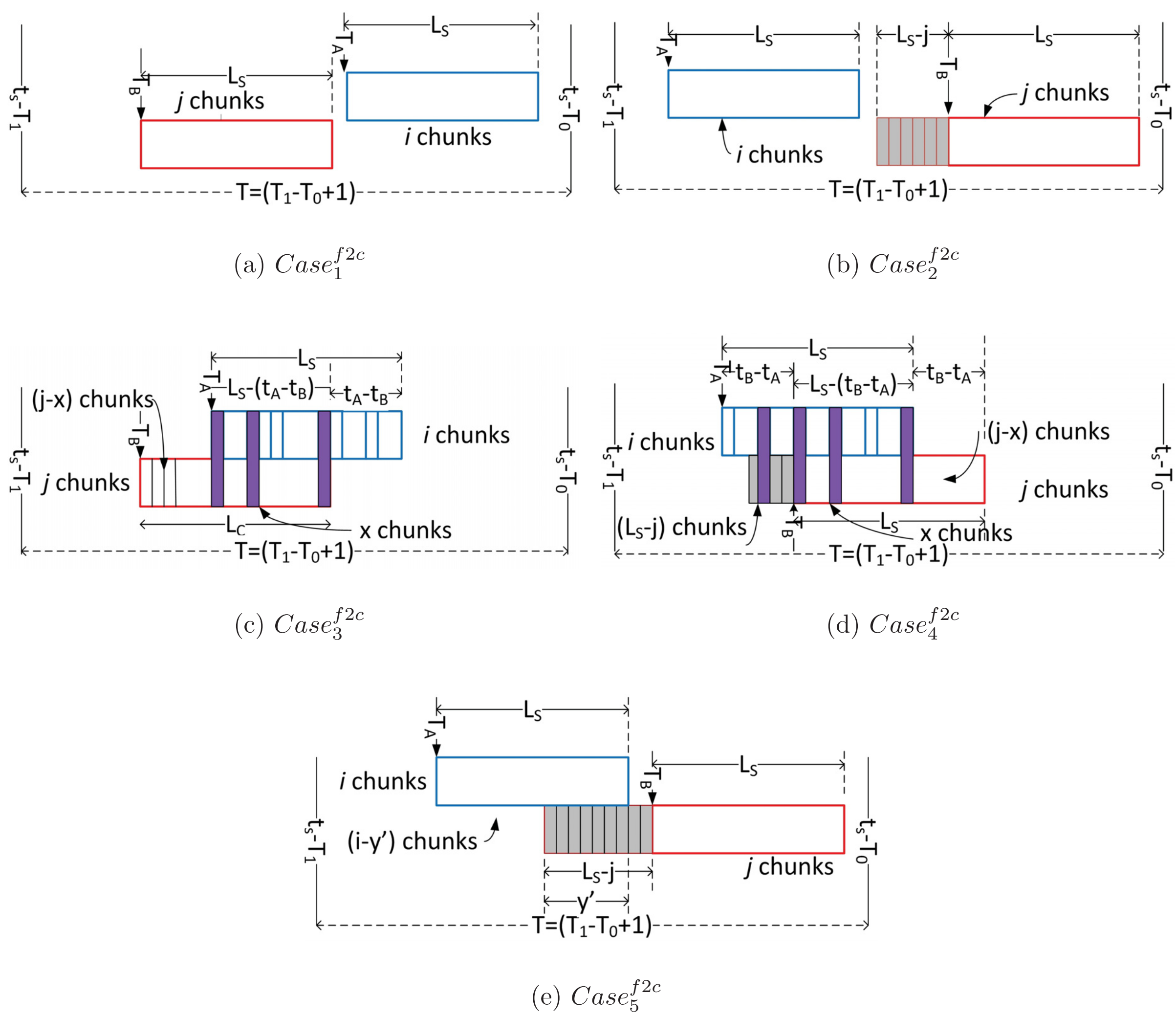
We denote by $f_k^{f2c}(i,j)$ the $f^{f2c}(i,j)$ probability in Case$_k^{f2c}$, where $k = 1...5$. As the $f_k^{f2c}(i,j)$ probabilities are mutually exclusive:

$$f^{f2c}(i,j) = f_1^{f2c}(i,j) + f_2^{f2c}(i,j) + f_3^{f2c}(i,j) + f_4^{f2c}(i,j) + f_5^{f2c}(i,j). \quad (177)$$

Computations of the $f_k^{f2c}(i,j)$ probabilities are as follows.
$Case_1^{f2c} : t_s - T_1 \leq t_A \leq t_D - L_s$
$Case_2^{f2c} : t_D + 2L_s - j \leq t_A \leq t_s - T_0$
$Case_3^{f2c} : t_D - L_s + 1 \leq t_A \leq t_D$
$Case_4^{f2c} : t_D + 1 \leq t_A \leq t_D + L_s - 1$

$Case_5^{f2c} : t_D + L_s \leq t_A \leq t_D + 2L_s - j - 1$

We, now calculate the probability, that free-riding peer D will not be interested in the chunks of its neighbor A. Let $f_k^{f2c}(i, j)$ be the probability for Case$_k^{f2c}$, where $k = 1...5$, such that, in the current time slot, free-riding peer D is not interested in the streaming chunks of its neighbor A under the condition of $Case_k^{f2c}$.
$Case_1^{f2c}[t_s - T_1 \leq t_A \leq t_D - L_s]$ (see Figure 36(a)):



(a) $Case_1^{f2c}$ (b) $Case_2^{f2c}$ (c) $Case_3^{f2c}$ (d) $Case_4^{f2c}$ (e) $Case_5^{f2c}$

Figure 36: Overlapping streaming chunks of free-riding peer D and one of its contributing neighbor A, in the Cooperative-to-contributing (f2c) cases

There is no overlapping of the respective buffers of contributing neighbor A and free-riding peer D. The playback pointer of peer D is already ahead of the playback pointer of neighbor A. It is obvious that peer D has no interest in the chunks of A irrespective of D's specific sub-stream.

$$f_1^{f2c}(i,j) = \sum_{t_D=t_s-T_1+L_s}^{t_s-T_0} \sum_{t_A=t_s-T_1}^{t_D-L_s} \frac{1}{T^2} \tag{178}$$

$Case_2^{f2c}[t_D + L_c + L_s - j \leq t_A \leq t_s - T_0]$ (see Figure 36(b)):
In this case, the playback pointer of the free-riding peer D plays behind the playback pointer of its contributing neighbor A and there is no overlapping of their respective buffers. As opposed to $Case_1^{f2c}$, buffer of the neighbor A contains more recent pieces than peer D. The lowest sequence number of the $(L_s - j)$ old chunks (in the buffer) of A is even higher than the highest sequence number of the chunks that D can store in its buffer in the current time slot. Hence, peer D, irrespective of its specific sub-stream, has no interest in the chunks of its neighbor A.

$$f_2^{f2c}(i,j) = \sum_{t_D=t_s-T_1}^{t_s-T_0-(2L_s-j)} \sum_{t_A=t_D+2L_s-j}^{t_s-T_0} \frac{1}{T^2} \tag{179}$$

$Case_3^{f2c}[t_D - L_s + 1 \leq t_A \leq t_D]$ (see Figure 36(c)):
Among the j useful chunks of neighbor A, x chunks are already present in the overlapping portion of the buffer of peer D. In other words, corresponding to the particular sub-stream of D, whatever chunks neighbor A has in the overlapping portion, peer D also has those chunks in the same portion. The remaining $(j - x)$ chunks of A are of no interest to D (as those chunks are already played by D).

$$f_3^{f2c}(i,j) = \sum_{t_D=t_s-T_1}^{t_s-T_0} \sum_{t_A=\max\{t_s-T_1,t_D-L_s+1\}}^{t_D} \sum_{x=\max\{0,i-(t_D-t_A)\}}^{\min\{(L_s-(t_D-t_A)),i,j\}} \left[\left(\frac{\binom{(t_D-t_A)}{j-x}\binom{L_s-(t_D-t_A)}{x}}{\binom{L_s}{j}}\right)\left(\frac{\binom{(L_s)-x}{i-x}}{\binom{L_s}{i}}\right)\right]\frac{1}{T^2} \tag{180}$$

$Case_4^{f2c}[t_D + 1 \leq t_A \leq t_D + L_c - 1]$ (see Figure 36(d)):
Among the j useful chunks, neighbor A has x chunks in the overlapping portion of its buffer and peer D already have all those x chunks. Moreover, peer D has all the $(L_s - j)$ old chunks of neighbor A corresponding to the sub-stream of D. Peer D is not interested in the

remaining $(j-x)$ chunks of A as those chunks are too new for the current time slot.

$$f_4^{f2c}(i,j) = \sum_{t_D=t_s-T_1}^{t_s-T_0} \sum_{t_A=\min\{t_s-T_0,t_D+1\}}^{\min\{t_s-T_0,(t_D+L_s-1)\}} \sum_{x=\max\{0,i-\min\{i,((t_A-t_D))\}\}}^{\min\{L_s-(t_A-t_D),i-\min\{(t_A-t_D),(L_s-j)\}\}} \left[\left(\frac{\binom{(t_A-t_D)}{j-x}\binom{L_s-(t_A-t_D)}{x}}{\binom{L_s}{j}}\right)\left(\frac{\binom{(L_s)-y}{i-y}}{\binom{L_s}{i}}\right)\right]\frac{1}{T^2} \tag{181}$$

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

$Case_5^{f2c}[t_D + L_s \leq t_A \leq t_D + 2L_s - j - 1]$ (see Figure 36(e)):
The free-riding peer D has i useful chunks in its buffer and the contributing peer A has j useful chunks and $L_s - j$ old chunks in its buffer. There is no overlapping of useful chunks of the respective buffers of D and A. However, useful chunks of peer D has overlapping with y' old chunks of peer A and peer D has all those y' chunks in its buffer. It is obvious that peer D will not be interested about the j useful chunks of peer A as those chunks are too new for peer D to buffer in the current time slot.

$$f_5^{f2c}(i,j) = \sum_{t_D=t_s-T_1}^{t_s-T_0-L_s} \sum_{t_A=t_D+L_c}^{\min\{t_s-T_0,[(t_D+2L_s-j-1]\}} \left[\frac{\binom{(L_c)-y'}{i-y'}}{\binom{L_s}{i}}\right]\frac{1}{T^2} \tag{182}$$

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

4.8.4 Probabilities of Requesting and Downloading Chunks for a Selfish Peer

We recall that, any random free-riding peer in this network has H_c fixed number of contributing neighbors and a buffer L_s of fixed length. It is also assumed that at any time slot if any randomly picked contributing peer (e.g. peer A) receives more than one request, u of

them (including the one from D) will be fulfilled randomly by peer A. We define $s_{\max}^{f2c}$, the maximum number of requests that an arbitrary free-riding peer (e.g., D) having i chunks in the current time slot can send to its neighbors as:
$s_{\max}^{f2c} = \min\{H_c, (L_s - i)\}$
where, $(L_s - i)$ is maximum number of chunks that peer D wants to download at that specific time slot.
Next, we define $F^{f2c}(H_c, i', k)$, as the probability that a randomly selected free-riding peer which has i useful chunks in its buffer, in a specific time slot, looking for $i' = (L_s - i)$, chunks in that time slot, sends k requests to its neighbors. $F^{f2c}(H_c, i', k)$ can be recursively calculated as follows:

$$F^{f2c}(H_c, i', k) = U_i^{f2c} F^{f2c}(h_c - 1, i' - 1, k - 1) + (1 - U_i^{f2c}) F^{f2c}(h_c - 1, i', k) \quad (183)$$

where, $i' = (L_s - i)$, $H_c \geq h_c \geq 0$, $\min\{i', h_c\} \geq k \geq 0$
The first component of the right hand side in (183) assumes the peer receives one of the requested chunks from one of the H_c contributing neighbors. Hence, in the second part, it is looking for one less chunk (i.e., $i' - 1$) from the remaining neighbors (i.e., $h_c - 1$) and sends one less request (i.e., $k - 1$).
The recurrence relation (183) has the following initial conditions:

$$F^{f2c}(h, i', k) = \begin{cases} 0 & \text{if } (h_c = 0 \text{ or } i' = 0) \text{ and } k > 0 \\ 1 & \text{if } (h_c = 0 \text{ or } i' = 0) \text{ and } k = 0. \end{cases}$$

Relation (183) can be derived by solving the following system of H_v equations:

$$\begin{aligned}
F^{f2c}(0,i',k) &= 0\\
F^{f2c}(1,i',k) &= U_i^{f2c}F^{f2c}(0,i'-1,k-1)\\
&\quad + (1-U_i^{f2c})(1-F^{f2c}(0,i',k))\\
F^{f2c}(2,i',k) &= U_i^{f2c}\times F^{f2c}(1,i'-1,k-1)\\
&\quad + (1-U_i^{f2c})\times F^{f2c}(1,i',k)\\
F^{f2c}(3,i',k) &= U_i^{f2c}\times F^{f2c}(2,i'-1,k-1)\\
&\quad + (1-U_i^{f2c})\times F^{f2c}(2,i',k)\\
\cdots &\quad \cdots\cdots\cdots\\
F^{f2c}(H_c-1,i',k) &= U_i^{f2c}\times F^{f2c}(H_c-2,i'-1,k-1)\\
&\quad + (1-U_i^{f2c})\times F^{f2c}(H_c-2,i',k)\\
F^{f2c}(H_c,i',k) &= U_i^{f2c}\times F^{f2c}(H_c-1,i'-1,k-1)\\
&\quad + (1-U_i^{f2c})\times F^{f2c}(H_c-1,i',k).
\end{aligned} \tag{184}$$

Next, we calculate the expected-value of $F^{f2c}(h_c,i',k)$ as $\overline{L}_{i'}^{f2c}$ and define $\overline{k}^{f2c}$ as the average number of requests that any arbitrary free-riding peer sends to its contributing neighbors at any given time slot.

$$\overline{L}_{i'}^{f2c} = \sum_{k=0}^{\min\{i',H_c\}} kF^{f2c}(h_c,i',k) \tag{185}$$

$$\overline{k}^{f2c} = \sum_{i=0}^{L_s}(P_i^{f2c}\overline{L}_i^{f2c}). \tag{186}$$

We define $\overline{X}^{f2c}$ as the average number of requests that a randomly selected contributing peer (e.g., peer A) receives from its neighbors in addition to the request received from free-riding peer D and Q^{f2c} as the probability that a randomly selected peer (e.g., peer A) arbitrarily fulfills u requests (i.e., including the one from peer D) among all the requests it received:

$$\overline{X}^{f2c} = \frac{(H_v-1)\overline{k}^{f2c}}{H_c}; \tag{187}$$

$$Q^{f2c} = \frac{u}{1+\overline{X}^{f2c}}. \tag{188}$$

Now, we compute the probability of forward transition in the discrete-time stochastic model shown in Figure 37 as follows:

We define $r^{f2c}_{i,n}$, as the probability that a free-riding peer, which has i useful chunks at a given time slot, downloads n chunks in the same time slot. It has a Binomial distribution with parameters k and Q^{f2c}. Hence: This is how it works, for the buffer, L_s

$$r^{f2c}_{i,n} = \sum_{k=n}^{\min\{(L_s-i),H\}} F^{f2c}(H,(L_s - i),k)\left(\binom{k}{n}(Q^{f2c})^n(1-Q^{f2c})^{k-n}\right) \quad (189)$$

Next, we define, M^{f2c}, probability that a randomly picked peer in the network with the buffer L_s and i uniformly distributed chunks, in its buffer in a specific time slot, can play (or track) a chunk (in the case of free-riding peer) in the same time slot. In fact this is the probability of reverse transition as shown in Figure 37.

Since we do not set priority to any of the chunks, the probability that we have the first chunk (for continuous playing or tracking) is:

$$M^{f2c}_i = \frac{i}{L_s} \qquad i = 0, 1, ..., L_s. \quad (190)$$

If the system is in state P^{f2c}_i, the probability the the system jumps to state P^{f2c}_{i+k} in the next time slot is given by the probability:

$$B^{f2c}_{i,i+k} = (1 - M^{f2c}_i)r_{i,k} + M^{f2c}_i r_{i,k+1} \quad (191)$$

The first term on the right hand side corresponds to the situation where the first chunk is missing (i.e., peer cannot playback in that time slot) and k chunks are downloaded. However, the second term represents a situation where the can peer play back in the current time slot and downloads $(k+1)$ chunks. When the system is in steady state, the probability distribution of any state in the model (shown in Figure 2) does not change with time and we get the following relation:

$$(B^{f2c}_{i+1,i}P^{f2c}_{i+1}) + \left(\sum_{k=0}^{\min\{H,i\}} (B^{f2c}_{i-k,i}P^{f2c}_{i-k})\right) - \left(\sum_{k=1}^{L_s-i}(B_{i,i+k}P^{f2c}_i)\right) - (B^{f2c}_{i,i-1}P^{f2c}_i) = 0. \quad (192)$$

It is difficult to find a closed form of solution for the relation in (192) in order to get the peer distribution $\left\{P_i^{f2c}\right\}$. However, it is possible to solve (192) numerically. Next, we define P_{cont}^{f2c}, as the probability that a randomly picked free-riding peer in the network that is keeping track of a certain live streaming content in a specific time slot, would be able to track the desired chunk at the same time slot. It can be expressed as follows:

$$P_{cont}^{f2c} = \sum_{i=0}^{L_s} P_i^{f2c} M_i^{f2c}. \tag{193}$$

Let d_i^{f2c} be the average download rate of a free-riding peer having i useful chunks in its buffer L_s in a specific time slot:

$$d_i^{f2c} = \sum_{k=0}^{\min\{H_c,(L_s-i)\}} \left(k r_{i,k}^{f2c}\right) \qquad i = 0, 1, ..., L_s. \tag{194}$$

The average download rate of a free-riding peer can be expressed as follows:

$$d^{f2c} = \sum_{i=0}^{L_s} \left(d_i^{f2c} P_i^{f2c}\right) \tag{195}$$

Figure 37 shows different states of A discrete-time Stochastic model for an arbitrarily selected free-riding peer in the HnH scheme based multi-channel live streaming system.

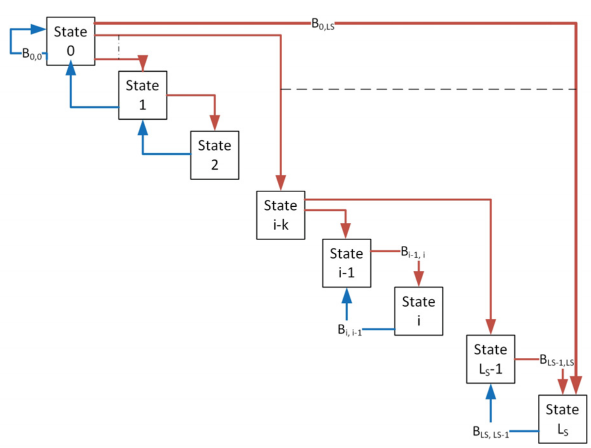


Figure 37: A discrete-time stochastic model for free-riding peers in a P2P live streaming system having buffer size L_s.

4.8.5 Selfish-to-Selfish (S2S) Peers' Interactions

In the F2F scenario, consider an arbitrary Free-Riding peer (e.g., peer D) and one of its neighbors (e.g., peer E) such that both with playback buffer of equal size, L_s. Suppose, at the current time slot, peer D has i useful chunks and neighbor E has j useful chunks in their respective buffers. t_D and t_E are the sequence numbers of the chunks to be played by the corresponding peer D and its neighbor E. Based on that, we calculate, U_i^{f2f}, the probability that peer D, while having i useful chunks in its buffer in a specific time slot, is interested in getting a chunk from a random Free-Riding neighbor, E. In order to calculate, U_i^{f2f}, let $f^{f2f}(i,j)$ be the probability that, at a given time slot, peer D with i useful chunks in its buffer is not interested in any of the j useful chunks of E. We then have:

$$U_i^{f2f} = \sum_{j=0}^{L_s} \left(1 - f^{f2f}(i,j)\right) P_j^{f2f}. \tag{196}$$

where P_j^{f2f} is the probability that a random Free-Riding peer in the network has j useful chunks. We calculate $f^{f2f}(i,j)$ for each of the following five mutually exclusive cases:
Case_1^{f2f} : $t_s - T_1 \leq t_B \leq t_A - L_s$
Case_2^{f2f} : $t_A + 2L_s - j \leq t_B \leq t_s - T_0$
Case_3^{f2f} : $t_A - L_s + 1 \leq t_B \leq t_A$
Case_4^{f2f} : $t_A + 1 \leq t_B \leq t_A + L_s - 1$
Case_5^{f2f} : $t_A + L_s \leq t_B \leq t_A + 2L_s - j - 1$. We denote by $f_k^{f2f}(i,j)$ be the probability that peer D is not interested in the useful pieces of peer E and these two peers are in Case_k^{f2f}, where $k = 1...5$. We assume that the playback pointer of a peer is uniformly distributed within the interval $[t_s - T_0, t_s - T_1]$ and $T = (T_1 - T_0 + 1)$. So, the probability that the playback pointer of peer D is in a given position is $1/T$. It is to be noted that, the definition of the probability of the playback pointer of any peer to be in a given position will be same for all the cases and the model will work with other distributions as well, however it will be more complicated
It follows:

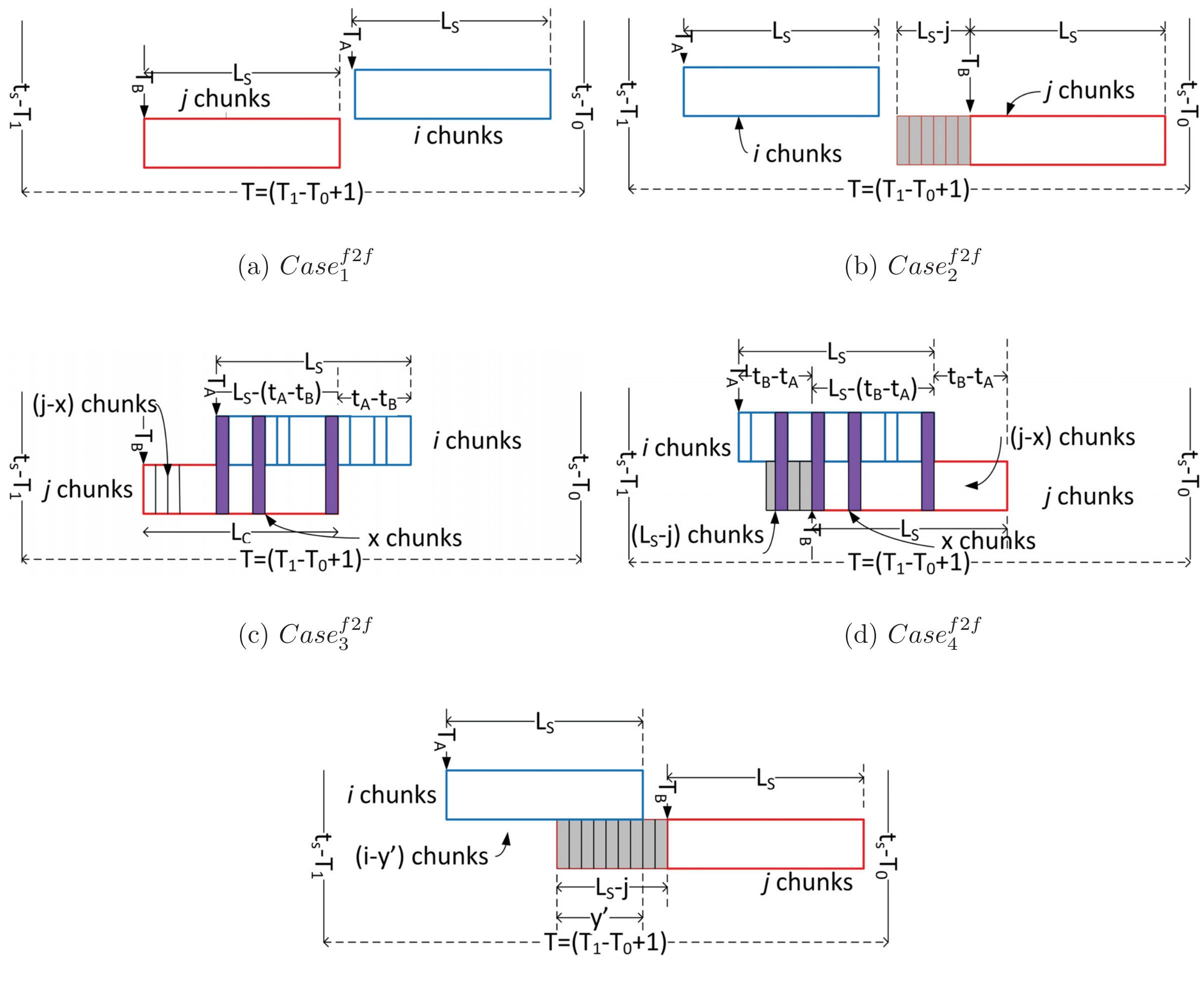
$$f_k^{f2f}(i,j) = \overline{f}_k^{f2f}(i,j) \cdot \frac{1}{T^2}. \tag{197}$$

Under these assumption, the definition of $f_k^{f2f}(i,j)$ will be same for all the cases in Case_k^{f2f}, where $k = 1...5$. As the $f_k^{f2f}(i,j)$ probabilities are mutually exclusive:

$$f^{f2f}(i,j) = f_1^{f2f}(i,j) + f_2^{f2f}(i,j) + f_3^{f2f}(i,j) + f_4^{f2f}(i,j) + f_5^{f2f}(i,j). \tag{198}$$

Computations of the $f_k^{f2f}(i,j)$ probabilities are as follows.

$Case_1^{f2f}: t_s - T_1 \le t_B \le t_A - L_s$ (see Figure 38(a)):



(a) $Case_1^{f2f}$ (b) $Case_2^{f2f}$

(c) $Case_3^{f2f}$ (d) $Case_4^{f2f}$

(e) $Case_5^{f2f}$

Figure 38: Overlapping of i useful streaming chunks of peer D with j useful chunks of one its random neighbor E in the Free-Riding-to-Free-Riding (f2f) cases

Peer D has i useful chunks and its random neighbor E has j useful chunks in their respective buffers. Moreover, those buffers have no content overlap. Peer D has already played all the j chunks of E and has no further interest in them. We suppose $\overline{f}_1^{f2f}(i,j)$, be the probability that D is not interested in E (i.e., irrespective of it the interval, T). Under the same assumption, the definition of the probability of the playback pointer of any peer to be in a given position will be same for all the subsequent cases. It follows:

$$f_1^{f2f}(i,j) = \sum_{t_A=t_s-T_1+L_s}^{t_s-T_0} \sum_{t_B=t_s-T_1}^{t_A-L_s} \frac{1}{T^2}. \tag{199}$$

In this case, $\overline{f}_1^{f2f}(i,j)$=1

$Case_2^{f2f}$: $t_A + 2L_s - j \leq t_B \leq t_s - T_0$ (see Figure 38(b)):

In this case, buffers of peer D and neighbor E have no content overlap and as opposed to $Case_1^{f2f}$, buffer of the neighbor E contains more recent pieces than peer D. The lowest sequence number of the $(L_s - j)$ old chunks (residing in the buffer) of peer E is even higher than the highest sequence number of the chunks that D can store in its buffer in the current time slot. Hence, peer D has no interest in its neighbor E. It follows:

$$f_2^{f2f}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0-(2L_s-j)} \sum_{t_B=t_A+2L_s-j}^{t_s-T_0} \frac{1}{T^2}. \tag{200}$$

$Case_3^{f2f}[t_A - L_s + 1 \leq t_B \leq t_A]$ (see Figure 38(c)):

Among the j useful chunks of neighbor E, x chunks are already present in the overlapping portion of the buffer of peer D. If D is not interested in E, then whatever chunks neighbor E has in the overlapping portion, peer D already has those chunks. The remaining $(j - x)$ chunks of E are of no interest to D (as those chunks are already played by D). It follows:

$$f_3^{f2f}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_B=\max\{t_s-T_1,t_A-L_s+1\}}^{t_A} \sum_{x=\max\{0,j-(t_A-t_B)\}}^{\min\{L_s-(t_A-t_B),i,j\}} \left[\left(\frac{\binom{t_A-t_B}{j-x}\binom{L_s-(t_A-t_B)}{x}}{\binom{L_s}{j}}\right)\left(\frac{\binom{L_s-x}{i-x}}{\binom{L_s}{i}}\right)\right]\frac{1}{T^2}. \tag{201}$$

In this case, $\overline{f}_3^{f2f}(i,j)$ is calculated inside the square braces. The first part of $\overline{f}_3^{f2f}(i,j)$ denotes the probability that neighbor E has j chunks in its buffer and among these j chunks, x chunks are in the portion which is overlapped with the buffer of peer D. The second part denotes that peer D has i chunks where these i chunks include all the x chunks which are in the buffer of neighbor E. Under the same assumption, $\overline{f}_k^{f2f}(i,j)$ will be calculated in the same manner for all the subsequent cases.

$Case_4^{f2f}$: $t_A + 1 \leq t_B \leq t_A + L_s - 1$ (see Figure 38(d)):

Neighbor E has x (out of j) useful chunks in the overlapping portion, however, peer D already has all the x chunks in its buffer. Moreover, peer D has all the $(L_s - j)$ old chunks of E. The remaining $(j-x)$ chunks of E are of no interest to D (as those chunks are too new for it for the current time slot). We then have:

$$f_4^{f2f}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0} \sum_{t_B=\min\{t_s-T_0,t_A+1\}}^{\min\{t_s-T_0,t_A+L_s-1\}} \sum_{x=\max\{0,j-(t_B-t_A)\}}^{\min\{L_s-(t_B-t_A),i-\min\{(t_B-t_A),(L_s-j)\},j\}} \left[\left(\frac{\binom{t_B-t_A}{j-x}\binom{L_s-(t_B-t_A)}{x}}{\binom{L_s}{j}}\right)\left(\frac{\binom{L_s-y}{i-y}}{\binom{L_s}{i}}\right)\right]\frac{1}{T^2}. \quad (202)$$

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

$Case_5^{f2f} : t_A + L_s \le t_B \le t_A + 2L_s - j - 1]$(see Figure 38(e)):

In this case, there is a overlapping between the i useful chunks of peer D and y' (among $(L_s - j)$) old chunks of neighbor E. Peer D is not interested in the j useful chunks of neighbor E (as those are too new for D in the current time slot). We have:

$$f_5^{f2f}(i,j) = \sum_{t_A=t_s-T_1}^{t_s-T_0-L_s} \sum_{t_B=t_A+L_s}^{\min\{t_s-T_0,[(t_A+L_s)+L_s-j-1]\}} \left[\frac{\binom{L_s-y'}{i-y'}}{\binom{L_s}{i}}\right]\frac{1}{T^2}. \quad (203)$$

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

Probabilities of requesting and downloading chunks for a cooperating peer in the f2f scenario

We recall that, any random peer in this network has H_c fixed number of Free-Riding neighbors, H_f fixed number of free-riding neighbors and a buffer L_s of fixed length. It is also assumed that at any time slot if any randomly picked peer (e.g. peer D) receives more than one request, u of them (including the one from D) will be fulfilled randomly by peer E. We define $s_{\max}^{f2f}$, the maximum number of requests that an arbitrary Free-Riding peer (e.g., D) having i chunks in the current time slot can send to its neighbors as:

$s_{\max}^{f2f} = \min\{H, (L_s - i)\}$

where, $(L_s - i)$ is maximum number of chunks that peer D wants to download at that specific time slot and $H = (H_c + H_f)$.

Next, we define $F^{f2f}(H, i', k)$, as the probability that a randomly selected Free-Riding peer which has i useful chunks in its buffer, in a specific time slot, looking for $i' = (L_s - i)$, chunks in that time slot, sends k requests to its neighbors. $F^{f2f}(H, i', k)$ can be recursively calculated as follows:

$$F^{f2f}(H, i', k) = U_i^{f2f} F^{f2f}(h-1, i'-1, k-1) + (1 - U_i^{f2f}) F^{f2f}(h-1, i', k) \quad (204)$$

where, $i' = (L_s - i)$, $H \geq h \geq 0$, $\min\{i', h\} \geq k \geq 0$ and $L_s \geq i \geq 0$,
The first component of the right hand side in (204) assumes the peer receives one of the requested chunks from one of the H viewing neighbors. Hence, in the second part, it is looking for one less chunk (i.e., $i' - 1$) from the remaining neighbors (i.e., $h - 1$) and sends one less request (i.e., $k - 1$).
The recurrence relation (204) has the following initial conditions:

$$F^{f2f}(H, i', k) = \begin{cases} 0 & \text{if } (h = 0 \text{ or } i' = 0) \text{ and } k > 0 \\ 1 & \text{if } (h = 0 \text{ or } i' = 0) \text{ and } k = 0. \end{cases}$$

Relation (204) can be derived by solving the following system of H_v equations:

$$\begin{aligned}
F^{f2f}(0, i', k) &= 0 \\
F^{f2f}(1, i', k) &= U_i^{f2f} F^{f2f}(0, i'-1, k-1) \\
&\quad + (1 - U_i^{f2f})(1 - F^{f2f}(0, i', k)) \\
F^{f2f}(2, i', k) &= U_i^{f2f} \times F^{f2f}(1, i'-1, k-1) \\
&\quad + (1 - U_i^{f2f}) \times F^{f2f}(1, i', k) \\
F^{f2f}(3, i', k) &= U_i^{f2f} \times F^{f2f}(2, i'-1, k-1) \\
&\quad + (1 - U_i^{f2f}) \times F^{f2f}(2, i', k) \\
\cdots &\quad \cdots\cdots\cdots \\
F^{f2f}(H-1, i', k) &= U_i^{f2f} \times F^{f2f}(H-2, i'-1, k-1) \\
&\quad + (1 - U_i^{f2f}) \times F^{f2f}(H-2, i', k) \\
F^{f2f}(H, i', k) &= U_i^{f2f} \times F^{f2f}(H-1, i'-1, k-1) \\
&\quad + (1 - U_i^{f2f}) \times F^{f2f}(H-1, i', k).
\end{aligned} \quad (205)$$

Next, we calculate the expected-value of $F^{f2f}(h, i', k)$ as $\overline{L}_{i'}^{f2f}$ and define $\overline{k}^{f2f}$ as the average number of requests that any arbitrary peer sends to its Free-Riding neighbors at any given time slot.

$$\overline{L}_i^{f2f} = \sum_{k=0}^{\min\{i',H\}} kF^{f2f}(h, i', k) \tag{206}$$

$$\overline{k}^{f2f} = \sum_{i=0}^{L_s} P_i^{f2f} \overline{L}_i^{f2f}. \tag{207}$$

We define $\overline{X}^{f2f}$ as the average number of requests that a randomly selected Free-Riding peer (e.g., peer D) receives from its neighbors in addition to the request received from free-riding peer D :

$$\overline{X}^{f2f} = \frac{(H-1)\overline{k}^{f2f}}{H}\ ; \tag{208}$$

and Q^{f2f} as the probability that a randomly selected peer (e.g., peer D) arbitrarily fulfills u requests (i.e., including the one from peer D) among all the requests it received:

$$Q^{f2f} = \frac{u}{1 + \overline{X}^{f2f}}. \tag{209}$$

4.8.6 Probabilities of Requesting and Downloading Chunks for a Contributing Peer with Incentive

We now assume that a peer will have both contributing and free-riding neighbors at the same time. Now, we define $F^{inc}(H_c, H_f, i', k_c, k_f)$, as the probability that a randomly selected peer (e.g., peer A) having i useful chunks in its buffer, in a specific time slot, looking for $i' = (L_s - i)$, chunks in that time slot, sends k_c requests to its H_c contributing neighbors and k_f requests to its H_f cooperative neighbors.

$F^{inc}(H_c, H_f, i', k_c, k_f)$, can be calculated from the following two recursive expressions:

$$F^{inc}(h_c, h_f, i', k_c, k_f) = U_i^{c2c} F(h_c - 1, h_f, i' - 1, k_c - 1, k_f) + (1 - U_i^{c2c}) F(h_c - 1, h_f, i', k_c, k_f) \tag{210}$$

and

$$F^{inc}(0, h_f, i'_f, 0, k_f) = U_i^{c2f} F(0, h_f - 1, i'_f - 1, 0, k_f - 1) + (1 - U_i^{c2f}) F(0, h_f - 1, i'_f, 0, k_f) \tag{211}$$

where, $i' = (L_s - i)$, $H_c \geq h_c \geq 0$, $H_f \geq h_f \geq 0$, $\min\{i', (H_c + H_f)\} \geq (k_c + k_f) \geq 0$, $\min\{i', H_c\} \geq k_c \geq 0$, $\min\{(i' - k_c), H_f\} \geq k_f \geq 0$ and $L_s \geq i \geq 0$, $i'_f = (L_s - i)$ such that

$h_c = 0$ and $k_c = 0$
The first component of the right hand side in (210) assumes the peer receives one of the requested chunks from one of the h_c contributing neighbors. Hence, in the second part, it is looking for one less chunk (i.e., $i' - 1$) from the remaining contributing neighbors (i.e., $h_c - 1$) and sends one less request (i.e., $k_c - 1$). The first and the second portion of (211) can be explained similar to (210).
The recursive equation (149) has the following initial conditions:
$F^{inc}(H_c, H_f, i', k_c, k_f) =$

$$\begin{cases} 1 & \text{if } ((h_c + h_f) ori' = 0) \; \& \; k_c = k_v = 0 \\ 0 & \text{if } ((h_c + h_f) ori' = 0) \; \& \; k_c + k_v > 0 \end{cases}$$

Next, we calculate the expected-value of $F(h_c, h_f, i', k_c, k_f)$ as $\overline{L}_{i'}$ and define $\overline{k}^{inc}$ as the average number of requests that any randomly selected peer sends to its neighbors at any given time slot.

$$\overline{L}_i^{inc} = \sum_{k_c=0}^{k_c^{max}} \sum_{k_f=0}^{\min\left\{(i'-k_c), H_f\right\}} (k_c + k_f) F(H_c, H_f, i', k_c, k_f) \tag{212}$$

$$\overline{k}^{inc} = \sum_{i=0}^{L_s} (P_i \overline{L}_i^{inc}) \tag{213}$$

where, $k_c^{max} = \min\{i', H_c\}$ We define $\overline{X}^{inc}$ as the average number of requests that any randomly selected peer in the network(e.g., peer B) receives from its neighbors in addition to the request received from peer A

$$\overline{X}^{inc} = \frac{(H_c - 1 + H_f)\overline{k}^{inc}}{(H_c + H_f)} \tag{214}$$

Next, we define the followings:
(i) Q^{c2c} as the probability that a randomly selected contributing peer (e.g., peer B) arbitrarily fulfills u requests of its contributing neighbors (i.e., including the one from peer A) among all the requests it received:

$$Q^{c2c} = \frac{u}{(1 - P_f)(1 + \overline{X}^{inc}) + P_f(1 + \overline{X}^{inc})q_f} \tag{215}$$

(ii) Q^{f2c} as the probability that a randomly selected contributing peer (e.g., peer B) arbitrarily fulfills u requests for its free-riding neighbors among all the requests it received:

$$Q^{f2c} = \frac{uq_f}{(1-P_f)(1+\overline{X}^{inc}) + P_f(1+\overline{X}^{inc})q_f} \tag{216}$$

$$Q^{c2f} = \frac{uq_f}{(1-P_f)(1+\overline{X}^{inc}) + P_f(1+\overline{X}^{inc})q_f} \tag{217}$$

(iv) Q^{f2f} as the probability that a randomly selected free-riding peer (e.g., peer D) arbitrarily fulfills u requests of its free-Riding neighbors(i.e., including the one from peer A) among all the requests it received:

$$Q^{f2f} = \frac{u(q_f)^2}{(1-P_f)(1+\overline{X}^{inc}) + P_f(1+\overline{X}^{inc})q_f} \tag{218}$$

Now, we compute the probability of forward transition for a contributing peer in the discrete-time stochastic model shown in Figure 2 as follows:
We define, $r^c_{i,n}$, probability that a contributing peer having i useful chunks at a given time slot, downloads n chunks in the same time slot. Hence, depending on the type of neighbors (i.e., contributing and free-riding), $r^c_{i,n}$ can be expressed as follows:

$$r^c_{i,n} = \sum_{k=n}^{\min\{(L_s-i),(H_c+H_f)\}} \sum_{k_f=0}^{\min\{k,H_f\}} F^{inc}(H_c, H_f, (L_s-i), k_c, k_f) \sum_{l=0}^{\min\{n,k_f\}} \left[\binom{k_f}{l}(Q^{c2f})^l(1-Q^{c2f})^{(k_f-l)}\right] \left[\binom{k-k_f}{n-l}(Q^{c2c})^{(n-l)}(1-Q^{c2c})^{(k-k_f-n+l)}\right] \tag{219}$$

The first binomial distribution with parameters k_f, Q^{c2f}, $B(k_f, Q^{c2f})$ denotes that out of the k_f requests to the free-riding neighbors l of them have been granted for downloading a chunk. The second binomial distribution with parameters $(k-k_f), Q^{c2c}$, $B((k-k_f), Q^{c2c})$ denotes that out of the remaining $(k-k_f)$ requests to the contributing neighbors $(n-l)$ of them have been granted for downloading a chunk. Next, we compute the probability of forward transition for a free rider in the discrete-time stochastic model shown in Figure 2 as

follows:
We define, $r^f_{i,n}$, probability that a free rider having i useful chunks at a given time slot, downloads n chunks in the same time slot. Hence, depending on the type of neighbors (i.e., contributing and free-riding), $r^f_{i,n}$ can be expressed as follows:

$$r^f_{i,n} = \sum_{k=n}^{\min\left\{(L_s-i),(H_c+H_f)\right\}} \sum_{k_f=0}^{\min\left\{k,H_f\right\}} F^{inc}(H_c, H_f, (L_s-i), k_c, k_f) \sum_{l=0}^{\min\left\{n,k_f\right\}} \left[\binom{k_f}{l}(Q^{f2f})^l(1-Q^{f2f})^{(k_f-l)}\right] \left[\binom{k-k_f}{n-l}(Q^{f2c})^{(n-l)}(1-Q^{f2c})^{(k-k_f-n+l)}\right] \quad (220)$$

The first binomial distribution with parameters k_f, Q^{f2f}, $B(k_f, Q^{f2f})$ denotes that out of the k_f requests to the free-riding neighbors l of them have been granted for downloading a chunk. The second binomial distribution with parameters $(k-k_f), Q^{f2c}$, $B((k-k_f), Q^{f2c})$ denotes that out of the remaining $(k-k_f)$ requests to the contributing neighbors $(n-l)$ of them have been granted for downloading a chunk. In this analysis, we assume that the desired chunks, which are scheduled to be downloaded in the current time slot, gets downloaded and stored in the buffer of that peer in the same time slot. We also assume that the streaming chunks are uniformly distributed across the buffer of a randomly selected peer in the network.
Next, we define, M^{inc}, probability that a randomly picked peer in the network with the buffer L_s and i uniformly distributed useful chunks, in its buffer in a specific time slot, can play a chunk in the same time slot. Since we do not set priority to any of the chunks, the probability that we have a chunk at the first place is:

$$M^{inc}_i = \frac{i}{L_s} \qquad i = 0, 1, ..., L_s. \quad (221)$$

If the system is in state P^{inc}_i, the probability the the system jumps to state P^{inc}_{i+k} in the next time slot is given by the probability:

$$B^{inc}_{i,i+k} = (1 - M^{inc}_i) r^{inc}_{i,k} + M^{inc}_i r^{inc}_{i,k+1} \quad (222)$$

The first term on the right hand side corresponds to the situation where the first chunk is missing (i.e., peer cannot playback in that time slot) and k chunks are downloaded. However, the second term represents a situation where the can peer play back in the current time slot and downloads $(k+1)$ chunks. When the system is in steady state, the probability distribution of any state in the model (shown in Figure 2) does not change with time and we get the following relation:

$$(B_{i+1,i}^{inc} P_{i+1}^{inc}) + \left(\sum_{k=0}^{\min\{H,i\}} (B_{i-k,i}^{inc} P_{i-k}^{inc}) \right) - \left(\sum_{k=1}^{L_s-i} (B_{i,i+k}^{inc} P_i^{inc}) \right) - (B_{i,i-1}^{inc} P_i^{inc}) = 0. \quad (223)$$

It is difficult to find a closed form of solution for equation 223 in order to get peer distribution $\{P_i^{inc}\}$. However, it is possible to solve (223) numerically. Next, we define P_{cont}^{inc}, 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 chunk at this time slot. It can be expressed as follows:

$$P_{cont}^{inc} = \sum_{i=0}^{L_s} P_i^{inc} M_i^{inc}. \quad (224)$$

Let d_i^{inc} be the average download rate of a peer that has i useful chunks in its buffer L_s in a specific time slot:

$$d_i^{inc} = \sum_{k=0}^{\min\{H,L_s-i\}} \left(k r_{i,k}^{inc} \right) \qquad i = 0, 1, ..., L_s. \quad (225)$$

Then the average download rate of a peer can be expressed as follows:

$$d^{inc} = \sum_{i=0}^{L_s} \left(d_i^{inc} P_i^{inc} \right). \quad (226)$$

4.9 Simulation and Numerical Results

Numerical Results here. In order to validate the proposed HnH scheme, we first get some numerical results (i.e., probability of continuity) without helping peers. Next, we introduce helping peers and then compare the probability of continuity with and without helping peers. In addition, in order to validate the proposed stochastic model, we compare the probability of continuity, P_{cont}, derived from the stochastic model with the probability of continuity obtained from our simulator. Moreover, we show improvement in downloading the chunks when helping peers are added. Finally, we compare the effective upload Bandwidth consumption, u_{eff} before and after the cooperating peers are added in the proposed HnH scheme.

4.9.1 Experimental Parameters

We assume that time is slotted and first set the normalized upload capacity, $u = 1$ chunk/time slot (later we will show cases where $u > 1$). In the experimental setup, a streaming file at the source is divided into $N = 500$ chunks, and $m = 4$ sub-streams. Each viewing peer in the HnH overlay is connected to a maximum of $H = 5$ viewing neighbors. Playback buffer size, L_s, of a viewing peer and the maximum playback delay, T, is considered in the range of: $2 \leq L_s \leq 15$ and $2 \leq T \leq 15$.

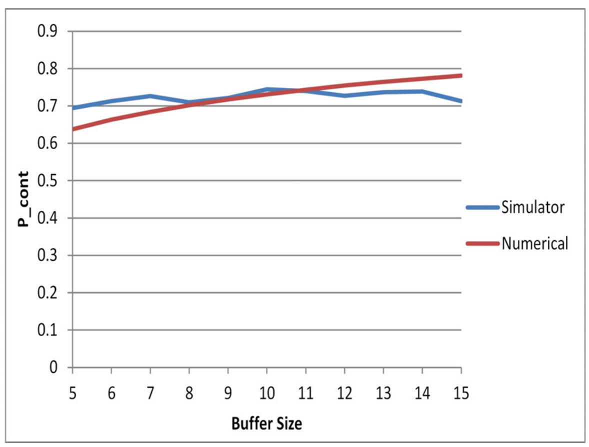


Figure 39: Comparing simulation and stochastic model results

Now, Figure 39 shows the comparison between the probability of continuity, P_{cont}, from the simulator and P_{cont} derived from the stochastic model. From this, we find that the stochastic model assumes realistic assumptions.

Figure 40, shows average download rate against the number of available (i.e., useful) chunks in the buffer before an after the cooperating peers are added. We find that the average download rate of peers have increased significantly.

4.9.2 Effect of Maximum Delay from the Source

Here we analyze the effect of maximum delay from the source on probability of continuity, P_{cont}. Figure 41, shows probability of continuity P_{cont} against maximum delay T for three

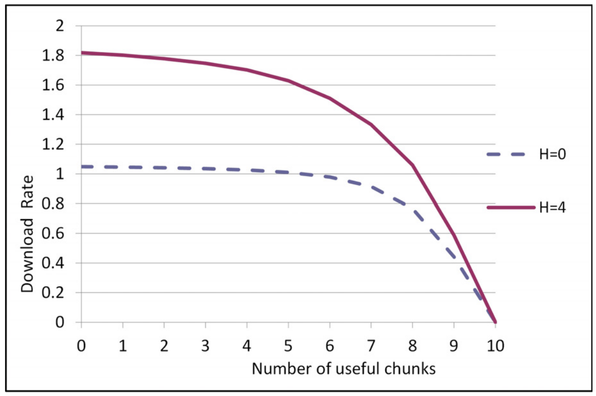


Figure 40: Improvement in downloading of chunks when helping peers are added

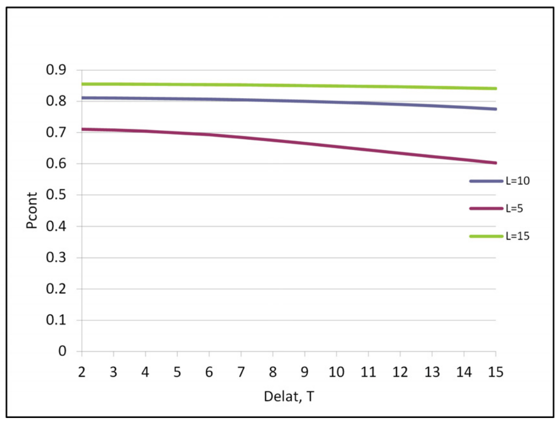


Figure 41: Probability of continuity (P_{cont}) decreases when maximum delay is increased

different values of buffer size $L_s = \{5, 10, 15\}$. The line at the bottom represents P_{cont} due to $L_s = 5$ and the one at the top due to $L_s = 15$.

4.9.3 Effect of Adding More Viewers from the Same Channel

In this section we analyze the effect on probability of continuity P_{cont} if we could have added more viewing peers from the same channel. However, in practice, we are unable to add more viewers to a DCSV channel. Figure 42, shows the effect of different number of viewers on the probability of continuity P_{cont} against buffer size L_s and maximum upload capacity $u_{\max} = 2.0$. The dotted line at the bottom represents number of viewer $v = 5$ and the one at the top $L_s = 15$.

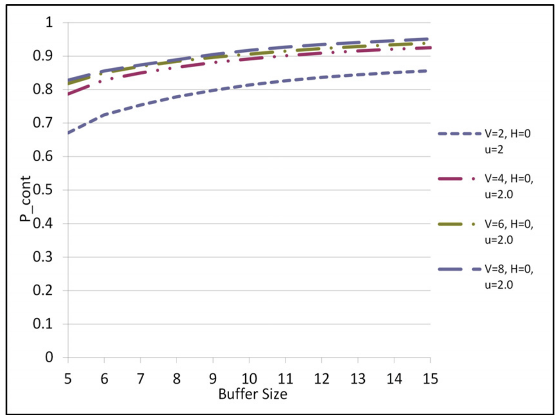


Figure 42: Effect of adding viewers from the same channel

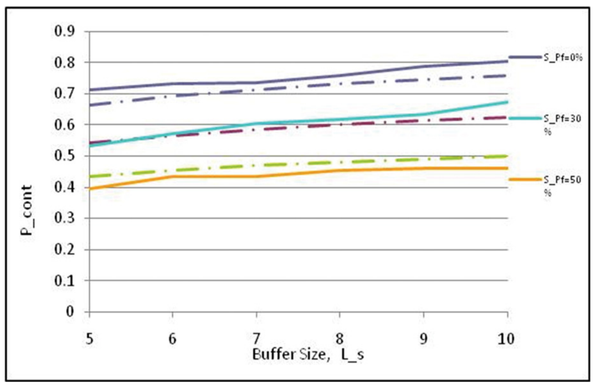


Figure 43: Probability of continuity (P_{cont}) increases when there is an incentive mechanism present

4.9.4 Effect of Incentive on the Selfish Peers

Here we analyze the effect of an incentive mechanism on the probability of continuity, P_{cont}. Figure 43, shows probability of continuity P_{cont} against buffer size L_s for different values of probability of free riding. The line at the bottom represents P_{cont} due to $Pf = 50\%$ and the one at the top due to $Pf = 0\%$ (i.e., no selfishness).

4.9.5 Effect of Maximum and Effective Upload Capacity of a Peer

We analyze here the effect of increasing the maximum upload capacity on probability of continuity, P_{cont}.

Figure 44 depicts the effect of increasing upload capacity u on the probability of continuity P_{cont} for different buffer size L_s with a number of viewers equal to 5.

Figure 45(a) compares the improvement of Probability of continuity P_{cont} when helping peers are added to small number of viewers with small maximum upload capacity, $u_{\max} = 1.0$

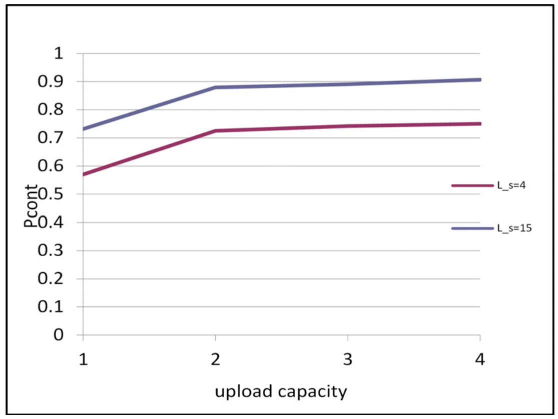
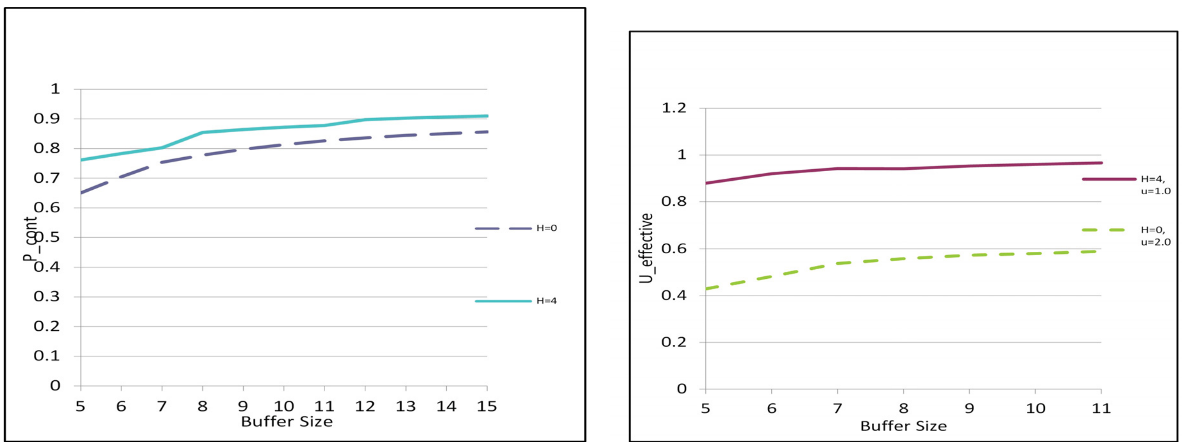


Figure 44: Probability of continuity vs. peers' upload capacity



(a) Probability of continuity P_{cont}

(b) Consumption of effective upload bandwidth, u_{eff}

Figure 45: Comparison of probability of continuity P_{cont} and consumption of effective upload bandwidth, u_{eff} before adding helping peers (upload rate $\rightsquigarrow u$=2.0) and after adding helping peers (upload rate $\rightsquigarrow u$=2.0)

with the case where upload capacity is increased to $u_{\max} = 2.0$ without adding helping peers. Figure 45(b) compares the corresponding consumption of effective upload BW, u_{eff}.

4.10 Conclusion and Future Work

Besides having free riders, we may find selfish or non-cooperating peers in some P2P live streaming systems, specially in the cooperation based systems. In our first research work, we proposed HnH scheme based P2P live streaming system. In this scheme, the cooperation

among the performance suffering peers from different DCSV channels is the key to enhancement and improvement of the overall channel viewing performance. However, if a peer from one DCSV channel decides not to cooperate with peers from another DCSV channel then this affects the overall performance of both DCSV channels participating in the HnH scheme. That is why these selfish peers are different from classical free riders. With respect to the performance problem due to the presence of selfish peers we had two main contribution: Firstly, we developed a discrete-time stochastic model in order to study the effect of selfish peers on the performance of the DCSV channels under HnH scheme. Our results show that presence of selfish peers certainly degrades the performance.
Secondly, we introduced a small incentive mechanism and modified the previous model. Then we find that presence of an incentive increase the performance of the DCSV channels. Future work in this topic may include: (i) Designing an efficient incentive mechanism for peers of DCSV channels under HnH scheme. (ii) Efficient selfish peer detection mechanism under HnH scheme.

Chapter 5

Conclusions and Future Work

In this thesis, we have studied the performance problem for P2P live streaming systems in terms of continuous playback and we provide insight for improving their performances. We have addressed two main aspects of performance problem: (a) content bottleneck problem of the peers of DCSV channels (i.e., short for Dedicated Channels used by a Small-numbered Viewers) and (b) presence of Free riders and selfish peers in the live streaming systems.

With respect to the content bottleneck problem for the DCSV channels, we have two main contributions:
Firstly, we propose a new cross-channel cooperation scheme, HnH (short for Hand-in-Hand), among the peers from different DCSV channels where every peer is very much likely to suffer from poor channel performance. Under the HnH scheme, the number of effective participants of a small channel increases and eventually improves the channel viewing performance.
Secondly, we have developed a discrete-time stochastic model in order to analyze the efficiency of the HnH scheme. Our proposed HnH scheme relies on natural incentive for cooperation among the performance deprived peers of DCSV channels who are assumed to be naturally interested to help each other for a better performance.
Future work of this topic may include: (i) Devising an efficient mechanism to gather two or more DCSV channels to form a HnH cooperation formation. (ii) Designing an efficient content retrieval mechanism for the DCSV channels under the HnH scheme.

Free riders are the peers who only want to download and watch the streaming content from their neighboring peers but are unwilling to upload any streaming content to their neighbors. The presence of Free riders impose obstacle to the stability of any live streaming system because of their consumption of bandwidth from the system without significant contribution. That is why, even channels with large participation may face playback performance degradation if good amount of Free riders are present in the system.

With respect to the performance problem due to the presence of the free riders, we have two main contributions:
Firstly, we have developed a discrete-time stochastic model in order to study the effect of Free riders on the performance of the channel. Our results show that presence of Free riders degrades the performance depending on the degree of Free riding.
Secondly, we have introduced a simple incentive mechanism and modified the previous model. We find that presence of an incentive mechanism increases the continuous playback performance of the system. Future work in this topic may include: (i) Designing an efficient incentive mechanism for peers in a live streaming system. (ii) Efficient Free rider detection mechanism.

Besides having Free riders, we may find selfish or non-cooperating peers in some P2P live streaming systems, specially in the cooperation based systems. In our first research work, we propose HnH scheme based P2P live streaming system. In this scheme, the cooperation among the performance suffering peers from different DCSV channels is the key to enhancement and improvement of the overall channel viewing performance. However, if a peer from one DCSV channel decides not to cooperate with peers from another DCSV channel then this non-cooperation affects the overall performance of both DCSV channels participating in the HnH scheme. These selfish peers are different from usual Free riders but important for the performance of HnH like cooperation based systems.

With respect to the performance problem due to the presence of selfish peers we have two main contributions:
Firstly, we have developed a discrete-time stochastic model in order to study the effect of selfish peers on the performance of the DCSV channels under HnH scheme. Our results show that presence of selfish peers certainly degrades the performance.
Secondly, we introduce a simple incentive mechanism and modify the previous stochastic model for HnH. We find that presence of an incentive mechanism increase the performance of the DCSV channels under HnH like cooperation scheme. Future work in this topic may include: (i) Designing an efficient incentive mechanism for peers of DCSV channels under HnH scheme. (ii) Efficient selfish peer detection mechanism under HnH scheme.

Bibliography

[1] http://www.napster.com.

[2] http://www.coolstreaming.us/.

[3] http://www.pptv.com/en/.

[4] http://www.sopcast.com/.

[5] C. Liang and Y. Liu, ViVUD: Virtual Server Cluster based View-Upload Decoupling for Multi-Channel P2P Video Streaming Systems, in: IEEE Global Telecommunications Conference - GLOBECOM, 2010, pp. 1–5.

[6] D. Qiu and R. Srikant, Modeling and performance analysis of BitTorrent-like peer-to-peer networks, in: SIGCOMM, 2004, pp. 1–11.

[7] Y. Zhou and D. Chiu and J.C. Lui., A Simple Model for Analysis and Design of P2P Streaming Algorithms, in: IEEE International Conference on Network Protocols - ICNP, 2007, pp. 1–10.

[8] Y. Zhou, D. Chiu, J. Lui, A simple model for chunk-scheduling strategies in P2P streaming, IEEE/ACM Transactions on Networking 19 (1) (2011) 42–54.

[9] R. Kumar, Y. Liu, K. Ross, Stochastic fluid theory for P2P streaming systems, in: IEEE Annual Joint Conference of the IEEE Computer and Communications Societies - INFOCOM, 2007, pp. 919 – 927.

[10] L. Massoulie, A. Twigg, C. Gkantsidis and P. Rodriguez, Randomized Decentralized Broadcasting Algorithms, in: IEEE Annual Joint Conference of the IEEE Computer and Communications Societies - INFOCOM, 2007, pp. 1073–1081.

[11] S. Liu and R. Zhang-Shen and W. Jiang and J. Rexford and M. Chiang, Performance Bounds for Peer-Assisted Live Streaming, in: ACM SIGMETRICS, 2008, pp. 313–324.

[12] C. Wu and B. Li and S. Zhao, Multi-channel Live P2P Streaming: Refocusing on Servers, in: IEEE Annual Joint Conference of the IEEE Computer and Communications Societies - INFOCOM, 2008, pp. 1355 – 1363.

[13] X. Jin, Y.-K. Kwok, Coercion builds cooperation in dynamic and heterogeneous p2p live streaming networks, Computer Networks. (2015) 1–18.

[14] Z. Kotevski, P. Mitrevski, Hybrid fluid modeling approach for performance analysis of P2P live video streaming systems, Peer-to-Peer Networking and Applications 7 (2014) 410 – 426.

[15] J. Zhang, W. Xing, Y. Wang, D. Lu, Modeling and performance analysis of pull-based live streaming schemes in peer-to-peer network, Computer Communications 40 (2014) 22 – 32.

[16] Y. Chen, C. Chen, D. M. Chiu, Performance modeling and evaluation of peer-to-peer live streaming systems under flash crowds, IEEE/ACM Transactions on Networking 22 (14) (2014) 518–526.

[17] B. Saleh, D. Qiu, Performance analysis of network-coding-based P2P live streaming systems, IEEE/ACM Transactions on Networking.

[18] D. Wu, C. Liang, Y. Liu and K. Ross, View-Upload Decoupling: Redesigning multi-channel P2P live video systems with View-Upload Decoupling, Computer Networks. (2010) 2007 2018.

[19] S. Liu and M. Chen and S. Sengupta and M. Chiang and J. Li and P. Chou, P2P Streaming Capacity under Node Degree Bound, in: IEEE International Conference on Distributed Computing Systems - ICDCS, 2010, pp. 587–598.

[20] Y. Liu, On the minimum delay peer-to-peer video streaming: how realtime can it be?, in: 15th International Conference on Multimedia - MULTIMEDIA, 2007, pp. 127–136.

[21] I. Shahriar, D. Qiu, B. Jaumard, Analysis of hnh model for live streaming channels with a small number of viewers, in: P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2012.

[22] I. Shahriar, D. Qiu, B. Jaumard, Cooperative channel resource sharing in p2p networks, in: Proceedings of the Fifth Symposium on Information and Communication Technology (SoICT), 2014.

[23] E. Adar and B. Huberman, Free riding on gnutella, in: First Monday, Vol. 5, 2000, pp. 42 –68.

[24] D. Qiu and Z. Ma, A Novel Optimistic Unchoking Algorithm for BitTorrent, in: CCNC, 2009.

[25] Z. Liu, Y. Shen, K.W. Ross, S.S. Panwar and Y. Wang, Substream Trading: Towards an open P2P live streaming system, in: Network Protocols, 2008. ICNP 2008. IEEE International Conference on, 2008, pp. 94 –103.

[26] http://www.bittorrent.org.

[27] S.G.M. Koo and C.S.G.Lee, An incentive-compatible mechanism for efficient distribution of bulk contents on peer-to-peer networks, Telecommunication Systems 34 (2007) 81–90.

[28] A. Habib and J. Chuang, Incentive mechanism for peer-to-peer media streaming, in: Proceedings of International Workshop Quality of Service (IWQoS), 2004, pp. 171 – 180.

[29] C. Kumar, K. Altinkemer, P. De, A mechanism for pricing and resource allocation in peer-to-peer networks, Electronic Commerce Research and Application 10 (2011) 26–37.

[30] T. Wu, W. Lee, N. Guizani, T. Wang, Incentive mechanism for p2p file sharing based on social network and game theory, Journal of Network and Computer Applications 41 (3) (2014) 47–55.

[31] E. Altman, P. Nain, A. Shwartz, Y. Xu, Predicting the impact of measures against P2P networks: Transient behavior and phase transition, IEEE/ACM Transactions on Networking 21 (3) (2013) 935–949.

[32] W. T. O. Zhen Wei Zhao, Sameer Samarth, Modeling the effect of user interactions on mesh-based p2p vod streaming systems, ACM Transactions on Multimedia Computing (2013) 1–22.

[33] I. Shahriar, D. Qiu, B. Jaumard, Modeling of free riders in p2p live streaming systems, in: International Conference on Computing, Networking and Communications (ICNC), 2017.

[34] M.K.H. Yeung and Y.K. Kwok,, Game-Theoretic Scalable Peer-to-Peer Media Streaming, in: Proceedings of IEEE International Symposium on Parallel and Distributed Processing (IPDPS'08), 2008.

[35] I. Shahriar, D. Qiu, B. Jaumard, Modelling and performance analysis of hnh: A novel approach for combining live streaming channels with a small number of viewers, Submitted for publication.

[36] I. Shahriar, D. Qiu, B. Jaumard, Probability model for live streaming channels having free riders with and without incentive mechanism, Submitted for publication.

[37] I. Shahriar, D. Qiu, B. Jaumard, Effect of less motivated selfish peers on the cooperation among the dedicated channels having small number of viewers, Submitted for publication.