Journal of the Nigerian Association of Mathematical Physics
***Volume27*, (*July, 2014*), pp 495 – 502**

Modeling and Analysis of P2PTV Network with Playback Time Differences Between Viewers

Adamu Aminu

Umaru Musa Yar'adua University, Katsina, P.M.B 2218, Katsina State, Nigeria.

Abstract

Recently there have been massive deployments of P2P applications, i.e. P2P file distribution applications (e.g. BitTorrent, Gnutella, Napster e.t.c.), P2P audio and video streaming applications (e.g. TVUsee, PPLive, UUStream e.t.c.). P2P video streaming networks are popularly known as P2PTV networks. In this paper, a mathematical model and performance analysis for a P2PTV network with viewers having different playback points was conducted. In the model, interactions and data exchange process between users was described in terms of discrete Markov chain. In the model, however, two network characteristics were considered, namely user churn and playback time differences between users, and later the analysis of how these characteristics affect buffer occupancy and playback continuity was conducted.

Keywords:P2P Networks, Quality of Service, Downloading Strategy, Playback Continuity, Buffer Occupancy

1.0 Introduction

Nowadays a lot of services and applications are being provided through the Internet, and many of these services are related to the provision of stream data to multiple users simultaneously, example of these services are Video on Demand, online broadcast of News and sporting events e.t.c. Traditionally such services were mainly provided via client-server model, like in the case of CDN. However, the price of these servers and their maintenance cost are enormously expensive, and as more and more people are becoming connected to high speed Internet access, the demand for online stream services is high, thus enormous resources are required to support or to provide services to ever increasing population of viewers. In many cases to overcome these short comings associated to the client-server model in the provision of video services (real time and on Demand services) many networks use peer-to-peer technology (P2P technology) [1]. In modern P2P networks, each user acts as both client and server. Therefore, users not only consume the resources of the network, but they also provide additional resources to it, and this is the major advantage of P2P networks over client-server networks, where all the resources came only from a single server. In P2P networks the more the number of users connected to it, the more the resources the network will have, this property of P2P networks makes them to be functional and efficient at any combination or quantity of users getting services from the network [2 – 6]. P2P networks are widely use for the exchange of different sort of data, e.g. elastic or stream traffic. Traffic measurements have shown that, P2P traffic consumes more than 60% of the whole Internet traffic [1]. Compared to traditional client-server file sharing systems like FTP, WWW e.t.c., P2P file sharing networks have one big advantage namely scalability. The performance of traditional client-server model deteriorates rapidly as the number of clients increases, but in well-designed P2P network more users generally means better performance.

There are mainly two types of P2P networks, namely P2P streaming networks and P2P file sharing networks [5, 7]. The requirement in P2P file sharing networks is that, a user downloads the complete required file or document before he uses it. The basic idea in P2P architecture, consist of the followings: whole file or document is divided into small blocks of data elements, known as chunks, and users exchange chunks among themselves, if a user is missing some chunks of the requested file, he request and download those missing chunks from neighboring users in the network downloading the same file. There is no restriction on the time needed by a user need to have the complete file. Examples of these networks that were successfully deployed are BitTorrent, Napster, Gnutella e.t.c. The basic QoE parameter for these networks is the total time needed or taken to download the complete file.

In P2P streaming networks the opposite is the case, users download video chunks and store them in the playback buffer, while playing or watching the successfully downloaded chunks in their buffer, each chunk in P2P streaming networks have

Corresponding author: Adamu Aminu E-mail:aminu@mail.ru,Tel.: +2348031581658.

playback deadline, and any chunk that arrived at the user's buffer after its playback deadline is useless.
The main performance parameter in file sharing networks is how long does it take to download a single file (file download time). Whereas in P2P streaming networks, the performance measures are playback continuity, startup delay and probability of universal streaming.
Fluid models are usually used as a tool to analyze the performance of file sharing systems, while discrete models are used to analyze the streaming systems [8, 9, 10]. In this paper a model for the analysis of streaming network with playback delay among users is presented.

2.0 Model description

In this section, the analysis of video distribution in P2P streaming network with *N* users will be conducted. Assume that the network has a single server, which distribute a video stream to the users (viewers). As stated in the introduction, in P2P streaming networks the video stream is fragmented into video units or video blocks known as video chunks, users in the network exchange among themselves these video chunks while watching the video stream [11]. However each video chunk has its own corresponding playback time within which it must be played out, these playback times of video chunks is referred to as time slots, thus the video playback process in these types of networks is divided into time slots. Therefore, the length of a single time slot is equivalent to the playback time of one video chunk. In order to maintain a steady video playback or video playback continuity each user has buffer that can store M+1 video chunks, meaning that, the buffer is partitioned into cells, the cells are numbered from 0 through M. 0-cell is designed to store the newly produced video chunk coming from server, other *m*-cells, $m = 1 \cdots \mathrm{M}-1$, are designed to store those video chunks that a user has successfully downloaded from the network during past time slots or those video chunks that a user will download in the coming time slots, and the buffer M-cell is designed to store the oldest video chunk that will move out of the buffer to the video player for playback in the next time slot, subsequently the buffer can be viewed as a sliding window, which slides at the end of each time slot.

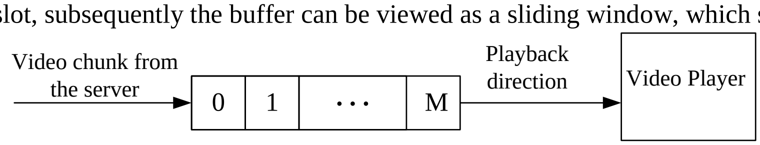

Figure.1a: Buffer and video player

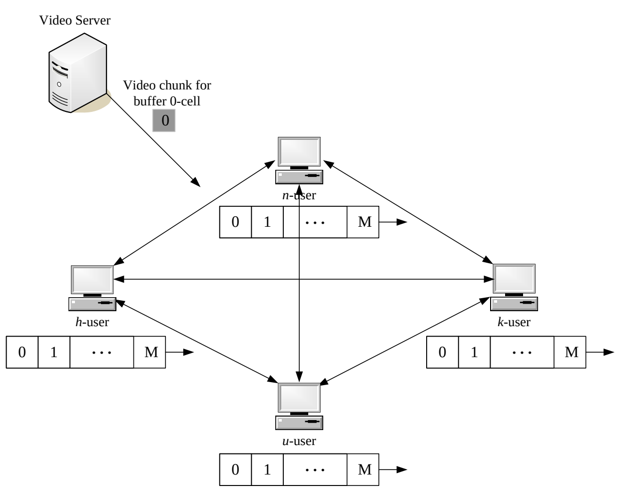

Figure.1b. Video exchange process

Now, the detail description of video playback process will be provided here. At the beginning of each time slot, the network server choose at random a user from the network and upload the freshly produced video chunk for the current time slot to its buffer 0-cell. The remaining N-1 users not chosen by the server in the current time slot will do the followings: if in the *n*-user's buffer there are empty cells, the *n*-user will choose at random another *h*-user (called a target user) from predefined group of his neighbors in order to fill up one of his empty cells, in other words, in order to download one of the missing video chunk from the target user. If the target user has one of the missing video chunk for one the corresponding empty cells in the *n*-user's buffer, so the download will be successful. If the target user has more than one of the missing video chunks for the corresponding empty cells in *n*-user's buffer, then the downloading strategy will define which of the video chunks *n*-user will download from his target user. Some of the widely used strategies are latest first strategy (LF strategy) and Greedy strategy. With LF strategy users during each time slot will seek to download the latest video chunk or the youngest video chunk among the missing ones; with Greedy strategy users tend to download the oldest video chunk among the missing ones.

If in the current time slot *n*-user has no empty cell in his buffer (has no missing video chunks in his buffer) or the *n*-user chose a target that does not have any of his missing video chunks, then *n*-user will not download a video chunk in the current time slot. At the end of each time slot video chunks in buffer cells will shift one step forward, i.e. video chunk in buffer M-cell will exit out of the buffer and move to the player for playback, the remaining video chunks in other buffer cells will shift one position to the right (towards the end of the buffer) to occupy the cell freed by its predecessor, and in this case buffer 0-cell will now be free to accommodate a new chunk from the server at the beginning of next time slot.

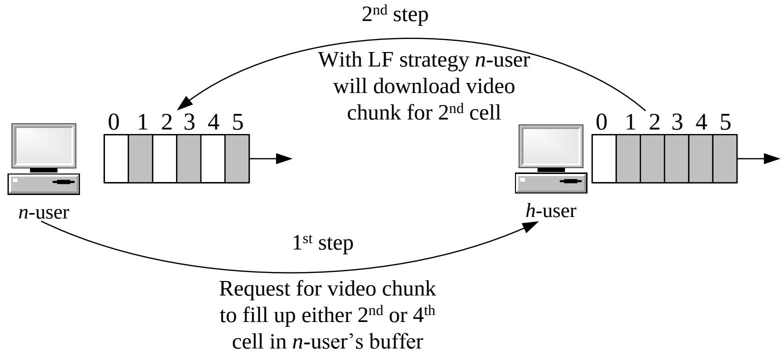

Figure.2. Latest First Strategy

It is assumed that each user may at some time leave the network and stop collaboration with other users, and likewise a user may join the network and start exchanging video chunks with other users in the network.

Next, a mathematical model for data exchange between users in the form of discrete Markov chain describing the buffer states of all users is presented. Here the model of user behavior, proposed in [5, 6], is extended by taking into account data transfer delays (playback time differences) that affect the video data exchange process between users.

3.0 Mathematical model

The following model for P2P streaming network will consider the playback lag (playback time differences) between users in the network. Because in the real life network some users are ahead of others in terms of playback position, i.e. for instance, consider a group of users watching the same video stream, some of the users in the same group may watch a moment before other members of the same group, this playback time differences between users is referred to as playback lag between users. Assume that in the network there are N subscribed users and a single server that distributes the video stream to the users. Users freely leave and join the network. Each user has a buffer for storing M-video chunks (Figure.3). A buffer cell may contain a video chunk and may be empty at any given time. Denote by $x_m(n)$ the state of *m*-cell in *n*-user's buffer, $m=\overline{1,\mathrm{M}}$ and $n=\overline{1,\mathrm{N}}$. $x_m(n)=1$ if *m*-cell is filled with a video chunk, otherwise $x_m(n)=0$. Subsequently the following vector defines the state of *n*-user's buffer $\mathbf{x}(n)=\left(x_m(n)\right)_{m=\overline{1,\mathrm{M}}}$. A user may at some times be in the network or out of the network, therefore user online indicator $a(n)$ define the state of *n*-user in the network, if $a(n)=1$ this means that the *n*-user is in the network otherwise $a(n)=0$.

Assume that the current time slot is t^l, at this time the video server produces a new video chunk and this current time slot is considered to be the server's playback time. Denote by $\tau_m^l(n)$ - a time slot associated to the *m*-cell in *n*-user's buffer in the current time slot, hence $\tau_m^l(n), m=\overline{0,\mathrm{M}}$ and $\tau_{\mathrm{M}}^l(n)<\tau_m^l(n)_{\overline{m=1,\mathrm{M}-1}}<\tau_0^l(n)\le t^l$.

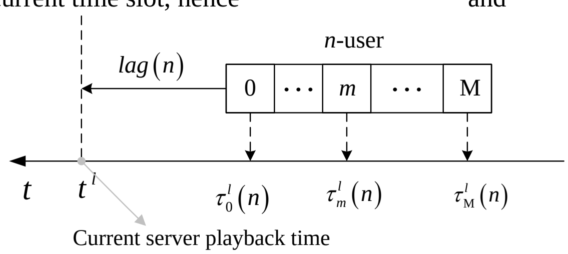

Figure 3. Playback lag illustration for *n*-user

Def. $lag(n) = t^l - \tau^l_{\mathrm{M}} - \mathrm{M} - 1$ is the playback lag of *n*-user at a time slot t^l.

The vector $\mathbf{z}(n) = (a(n), lag(n), \mathbf{x}(n))$ describe the state of *n*-user. Because we have N users in the network, therefore the matrix $\mathbf{X} = (x_m(n))_{m=\overline{0,\mathrm{M}},\, n=\overline{1,\mathrm{N}}}$, define the buffer state of all users in the network. Where $\dim \mathbf{X} = \mathrm{N}(\mathrm{M}+1)$. The state of the entire network is defined by $\mathbf{Z} = (\mathbf{a}, \mathbf{lag}, \mathbf{X})$, where vector $\mathbf{lag} = (lag(1), \cdots, lag(N))$ - contain the playback lag of all users in the network, and $\mathbf{a} = (a(1), \cdots, a(N))$ is an online indicator vector, $\mathbf{X}$ - buffer state matrix. Denote by $M^0(\mathbf{x}(n))$ and $M^1(\mathbf{x}(n))$ the set of empty and filled cells in *n*-user's buffer.

$$M^0(\mathbf{x}(n)) = \{m:\ x_m(n) = 0,\ m = \overline{1,\mathrm{M}}\},$$
$$M^1(\mathbf{x}(n)) = \{m:\ x_m(n) = 1,\ m = \overline{1,\mathrm{M}}\}, \qquad (1)$$

where $M^0(\mathbf{x}(n)) \cup M^1(\mathbf{x}(n)) = \{1, 2, ..., \mathrm{M}\}$.

Assume we have two groups of users in terms of playback lag, N^1 and N^2, users in the same group have equal playback lag, and $lag(h)_{\forall h \in \mathrm{N}^2} > lag(n)_{\forall n \in \mathrm{N}^1}$. Consider two users from different groups *h*-user from N^2 and *n*-user from N^1. Here there are two possible scenarios, either *n*-user chose *h*-user as a target user or *h*-user selected *n*-user as a target user.

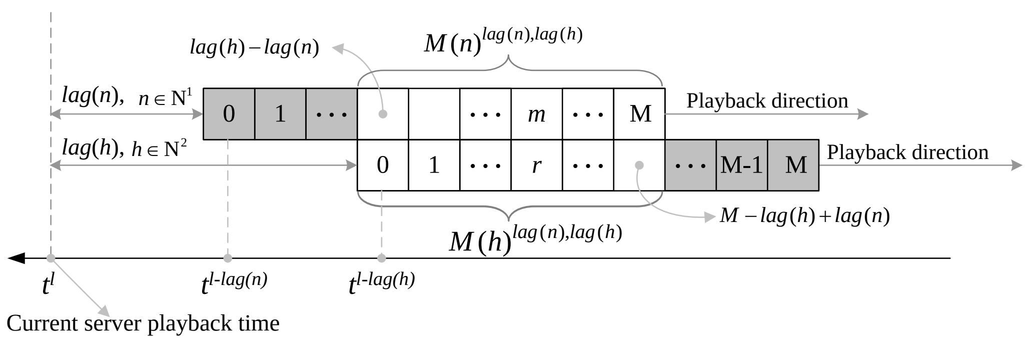

Figure.4.:The scheme for $M(n)^{lag(n),lag(h)}$ and $M(h)^{lag(n),lag(h)}$ sets

Now let's consider the first scenario, *n*-user chose *h*-user as a target user and $lag(n) \le lag(h)$, denote by $M(h)^{lag(n),lag(h)}$ the set of cells in *h*-user's buffer from which *n*-user could download a video chunk, where

$$M(h)^{lag(n),lag(h)} = \{0, 1, \cdots, \mathrm{M} - lag(h) + lag(n)\} \text{ if } lag(n) \le lag(h). \qquad (2)$$

For the case of *h*-user choosing *n*-user as a target user, we have $lag(h) > lag(n)$, then set $M(n)^{lag(n),lag(h)}$ will be the set of cells in *n*-user's buffer from which *h*-user could download a video chunk, where

$$M(n)^{lag(n),lag(h)} = \{lag(h) - lag(n), \cdots, \mathrm{M}\} \text{ if } lag(h) > lag(n). \qquad (3)$$

Now let's take the first scenario for further description. Assume *n*-user choose *h*-user as a target user, then $M^0(\mathbf{x}(n)) \cap M(h)^{lag(n),lag(h)}$ will be the set of cells in *n*-user buffer to which he can download video chunk from his target user, i.e. *h*-user. And the set $M^1(h) \cap M(n)^{lag(n),lag(h)}$ will be the set of filled cells in *h*-user's buffer, which he can upload to *n*-user. Due to playback lags one and the same data chunk in the buffers of users with different data transfer delays will be located in cells with different indexes. In order to establish a correspondence between these cells the following operation was used: $m = r - lag(n) + lag(h)$ (see Figure.4), where *m* is an index of buffer cell for *n*-user, and *r* is the corresponding index of buffer cell for *h*-user, $m \in M(n)^{lag(n),lag(h)}$ and $r \in M(h)^{lag(h),lag(n)}$. Thereby, the index $m_\delta(\mathbf{x}(n), \mathbf{x}(h), lag(n), lag(h))$ of *n*-user's buffer cell to which *n*-user can download a chunk from *h*-user is determined by the downloading strategy in use:

$$m_\delta(\mathbf{x}(n), \mathbf{x}(h), lag(n), lag(h)) = \min\Big\{\Big(M^0(\mathbf{x}(n)) \cap M(h)^{lag(n),lag(h)}\Big) \cap$$
$$\cap \Big\{m : m = r - lag(n) + lag(h), r \in \Big(M^1(\mathbf{x}(h)) \cap M(n)^{lag(n),lag(h)}\Big)\Big\}\Big\}. \quad (4)$$

For the shift of buffer content at the end of each time slot, the concept of shifting operator is introduced, with shifting operator if the *n*-user's buffer is found in the state $\mathbf{x}(n) = (x_0(n), x_1(n), \cdots, x_{\mathrm{M}}(n))$, with shifting operator the state will change to $S\mathbf{x}(n) = (0, x_0(n), \cdots, x_{\mathrm{M}-1}(n))$. Assume t_l be the shifting moment of buffer contents in the network. Because of the discrete nature of the model, it is assumed that if at the moment $t_l - 0$ a buffer is in state $\mathbf{x}(n)$ then at $t_l + 0$ will be at $S\mathbf{x}(n)$.

Users join and leave the network freely, based on the previous reason, assume that the joining and leaving events occurs only at the moment t_l. Let $a^l(n)$ be the user online indicator at the moment $t_l - 0$. If before the moment t_l a user was already in the network or joined the network at that moment, then $a^l(n) = 1$, otherwise $a^l(n) = 0$. A user join the network with probability $\alpha(n) = 0$ and leave the network with probability $\beta(n) = 0$, thus we have:

$$P\{a^{l+1}(n) = 1 \mid a^l(n) = 0\} = \alpha(n)$$
$$P\{a^{l+1}(n) = 0 \mid a^l(n) = 0\} = 1 - \alpha(n)$$
$$P\{a^{l+1}(n) = 0 \mid a^l(n) = 1\} = \beta(n)$$
$$P\{a^{l+1}(n) = 1 \mid a^l(n) = 1\} = 1 - \beta(n) \quad (5)$$

For simplicity assume that all users join and leave the network with equal probabilities, i.e. $\alpha(n) = \alpha$, $\beta(n) = \beta$, $n = \overline{1, \mathrm{N}}$. When a user leave the network, the corresponding raw in the matrix $\mathbf{X}$ will reset to zero vector, $\mathbf{x}(n) = \mathbf{0}$.

Let's examine the video distribution process and exact actions taken by the server and the users within the current time slot, $[t_l, t_{l+1})$:

a. Assume the current time slot is t_l, at this moment an offline user decides to join the network with probability α and online user decides to leave the network with probability β.

b. At the moment t_l, there will be a buffer content shift for all the users in the network, i.e.:

   i. Chunk in the buffer M-cell will exit out the buffer to player for playback
   ii. All other chunks in other buffer cells will be shifted one position to the right, i.e. towards the end of the buffer;
   iii. Buffer 0-cell will be emptied.

c. At the moment $t_l + 0$, server chooses one user randomly and uploads a chunk for the current time slot to his buffer 0-cell. If server has chosen *n*-user, then $x_0(i) = 1$ at the moment $t_{l+1} - 0$.

d. Each user (*n*-user), not chosen by the server, randomly chooses one of his neighbors as a target user (*h*-user), and according to download strategy δ tries to download of his missing chunks. If according to the strategy δ index $m_\delta(\mathbf{x}(n), \mathbf{x}(h), lag(n), lag(h))$ of a cell in *n*-user's buffer is chosen, then he will download that chunk from *h*-user. No actions are performed otherwise.

The state of the entire network at moment t^l can be denoted as $\mathbf{Z}^l = (\mathbf{a}^l, \mathbf{lag}^l, \mathbf{X}^l)$, subsequently the set $\{\mathbf{Z}^l\} := \{\mathbf{Z}^l, l \geq 0\}$ will form a Markov chain with a state space Z and one class of essential states $\tilde{Z}$, $\tilde{Z} \subset Z$.

Denote by $\pi^l(\mathbf{Z})$ the probability that Markov chain $\{\mathbf{Z}^l\}_{l\geq 0}$ is in state $\mathbf{Z}$ during the l^{th} time slot, i.e. $\pi^l(\mathbf{Z}) = P\{\mathbf{Z}^l = \mathbf{Z}\}$ and denote by $\Pi^{l,l+1}(\mathbf{Z},\mathbf{Y})$ the corresponding transition probability. The transition probability $\Pi^{l,l+1}(\mathbf{Z},\mathbf{Y})$ depends on $m_\delta\left(\mathbf{x}(n), \mathbf{x}(h), lag(n), lag(h)\right)$, i.e. on the strategy in use, on the playback lags and on α and β (user joining and leaving probabilities). The probability distribution $\pi^l(\mathbf{Z})$ satisfies the Kolmogorov-Chapman equations:

$$\pi^{l+1}(\mathbf{Y}) = \sum_{\mathbf{Z}\in Z} \pi^l(\mathbf{Z}) \Pi^{l,l+1}(\mathbf{Z},\mathbf{Y}), \mathbf{Y} \in Z, l \geq 0. \tag{6}$$

4.0 Playback continuity analysis

The basic QoE parameter in P2P streaming network is probability of playback continuity, which is a probability that a user is watching a stream without pauses during playback time, and this QoE parameter is denoted by V(n), and this probability correspond to the probability that M-cell in n-user's buffer is occupied with the corresponding chunk at the end of each time slot. To aid find this probability the following auxiliary function was introduced:

$$h_n^i(\mathbf{Z}) = \sum_{h=\overline{1,N} h\neq n, a^l(h)=1} \delta_{m_\delta(x(n),x(h),lag(n),lag(h)),i}, \mathbf{Z} \in Z$$

Where

$$\delta_{i,j} = \begin{cases} 1, & j = i, \\ 0, & j \neq i. \end{cases}$$

Denote by $\mathrm{N}(\mathbf{a}^l) = \sum_{j=\overline{1,N}} a^l(j)$ the number of online users when the network is in state $\mathbf{Z}^l = (\mathbf{a}^l, \mathbf{lag}^l, \mathbf{X}^l)$. Subsequently, the probability that the chunk needed by n-user to fill his i-buffer cell is available in the network during the l-th time slot is denoted by $Q_n^l(i)$. However, the probability $Q_n^l(i)$ depend on the downloading strategy δ, therefore this probability can be interpreted as the probability that n-user will select i-cell and successfully download a chunk from the target user during the l-th time slot. If $\mathrm{N}(\mathbf{a}^l) \geq 2$, the following formula can be obtained.

$$Q_n^l(0) = 0; \quad Q_n^l(i) = \sum_{\mathbf{Z}\in Z} \pi^l(\mathbf{Z}) \cdot \frac{h_n^i(\mathbf{Z})}{\mathrm{N}(\mathbf{a}^l) - 1}; \; i = \overline{1,\mathrm{M}}. \tag{7}$$

Denote by $p_0^l(n,i)$ and $p_1^l(n,i)$ the probability that n-user's i-buffer cell is empty and the probability that n-user's i-buffer cell is filled during the l-th time slot respectively. The following recursive relation for buffer cell state probabilities can be obtained:

$$p_1^l(n,0) = \frac{1}{N},$$

$$p_1^{l+1}(n,i+1) = p_1^l(n,i) \cdot (1-\beta) + p_0^l(n,i) \cdot (1-\beta) \cdot Q_n^l(i) + \alpha \cdot Q_n^l(i), \; i = \overline{0,\mathrm{M}-1}. \tag{8}$$

Assume that the equilibrium distribution of the Markov chain $\{\mathbf{Z}^l\}$ exists. Denote by $p_1(n,i) = \lim_{l\to\infty} p_1^l(n,i)$ the probability that n-user's i-buffer cell is filled with the corresponding video chunk, and by $p_0(n,i) = \lim_{l\to\infty} p_0^l(n,i)$ the probability that n-user's i-buffer cell is empty. From (8) we will have:

$$p_1(n,0) = \frac{1}{\mathrm{N}},$$

$$p_1(n,i+1) = p_1(n,i)\cdot(1-\beta) + p_0(n,i)\cdot(1-\beta)\cdot Q_n(i) + \alpha\cdot Q_n(i). \quad (9)$$

The probability of playback continuity has the following form:

$$V(n) = p_1(n, \mathrm{M}) \quad (10)$$

Hence the desired probability measure for the considered model is obtained.

I. Case Study

For the case study, consider a network with 300 subscribers, i.e. N=300, each user has a buffer with 40 positions (cells) for storing video chunks, M=40. Users have equal number of neighbors with whom they interact or exchange video chunks, the number of neighbors for each user is chosen to be 60. For simplicity of the analysis let's consider a network where all users are present in the network and do not leave the network, meaning we have $\alpha = 0$ and $\beta = 0$. Subscribers (users) are divided into three categories (groups) in terms of playback lag, assume that the playback lag of users in the same group is equal, $N = \bigcup_{i=1}^{3} N_i$, $lag(n) = lag(\hat{n})$, $n \neq \hat{n}$, $n, \hat{n} \in N_i$, $i = 1,2,3$.

Consider the case where the playback lag of the first group is zero and the playback lag of the second and third groups are 10 and 20 time slots respectively. The graphs in Figure 5 and in Figure 6 show that the group with the largest playback lag (the third group) will have the greatest probability of watching video stream without pauses during playback. This can be explained by the fact that any data chunk will be highly available among users from the first and second groups by the time it is requested by the users from the third group. These graphs don't lead to the conclusion which of two strategies is better. It is for further study.

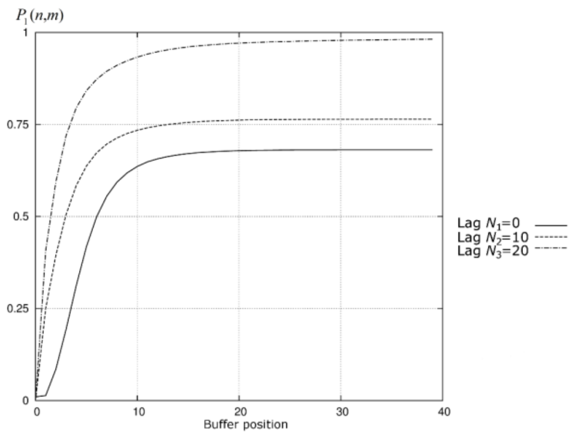

Figure.5: Buffer occupancy probability in case of LF strategy

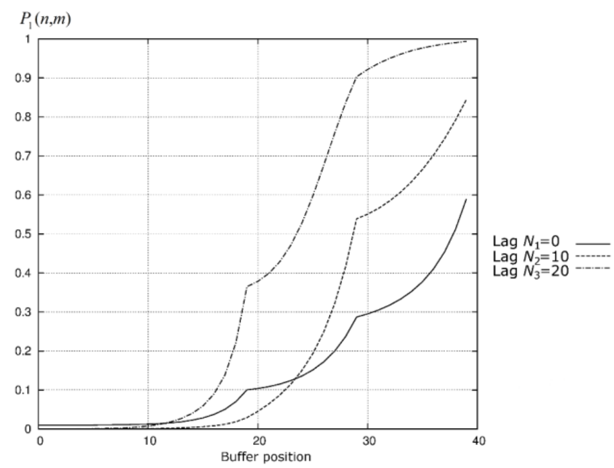

Figure.6. Buffer occupancy probability in case of Gr strategy

5.0 Conclusion

In this paper a model of data exchange between users of P2P live streaming network was developed in terms of discrete Markov chain, through which the formulas for the analysis of the system performance measures were obtained. In case when α=0 and β=0, the analysis of the network was conducted for two most popular downloading strategies, i.e. $\delta \in \{LF, Gr\}$. The direction for further study is to take into account different downloading rates of users in P2P network and to analyze both performance measures, i.e. universal streaming and playback continuity, within different modeling approach.

References

[1.] Setton E., Girod B. Peer-to-Peer Video Streaming // Springer. – 2007. - 150 p.

[2.] François Baccelli, Fabien Mathieu and Ilkka Norros. Performance of P2P networks with spatial interactions of peers // Networks and Telecommunications Networks, Systems and Services, Distributed Computing. Equipes-Projets GANG, TREC, Centre de recherche INRIA Paris, Rapport de recherche n° 7713 – August 2011. – P. 1-23.

[3.] Lei Guo, Songqing Chen, Zhen Xiao, Enhua Tan, Xiaoning Ding and Xiaodong Zhang. A performance study of BitTorrent-like peer-to-peer Systems // Proc. of the IEEE international conference on communication. Vol. 25, № 1, 2007. – P. 155-169.

[4.] Samuli Aalto, Manoj Bhusal. Effect of service time distribution on the performance of P2P file sharing // Special Assignment, Helsinki University of Technology, Department of Communications and Networking, S-38.3138 Networking Technology, 2008. – P. 1-27.

[5.] Adamu A., Gaidamaka Yu., Samuylov A. Analytical modeling of P2PTV network // Proc. of the 2nd International Congress on Ultra Modern Telecommunications and Control Systems (IEEE ICUMT 2010), Oct. 18-20, 2010. - Moscow, Russia. – P. 1115-1120.

[6.] Adamu, A., Gaidamaka, Yu., Samuylov, A. Discrete Markov Chain Model for Analyzing Probability Measures of P2P Streaming Network. Lecture Notes in Computer Science. Germany, Heidelberg: Springer. 2011. Vol. 6869. Pp. 428-439.

[7.] Srikant R and Qiu D. Modeling and performance analysis of BitTorrent-like peer-to-peer networks // Proc. of the ACM SIGCOMM Computer Communication Review, 2004. – P. 367-378.

[8.] Clevenot F. and Nain P. A Simple fluid model for the analysis of the squirrel peer-to-peer caching system // Proc. of the IEEE INFOCOM. – 2004. P. 1-10.

[9.] Dah M. Chiu, Bin Fan, Lui, J.C.S. Stochastic differential equation approach to model BitTorrent-like P2P systems // Proc. of the IEEE international conference on communication, 2006. – P. 915-920.

[10.] Kleinrock L. and Tewari S. Analytical model for bittorrent-based live video streaming // Proc. of the IEEE CCNC, 2007. – P. 976-980.

[11.] Xuemin Shen, Heather Yu, John Buford, Mursalin Akon. Handbook of Peer-to-Peer Networking // Springer. – 2010. - 1421 p