ISSN: 2350-0328
International Journal of AdvancedResearch in Science, Engineering and Technology
Vol. 6, Issue 8 , August 2018

Across Channel Resource Sharing (ACRS) Framework for P2P Live Streaming Network

Jamilu, B. A., Sani, Z. , Karmanje, A. R.

Department of Computer Science and Information Technology, Federal University Dutsin-Ma, Katsina State, Nigeria.

ABSTRACT: Our analysis of existing P2P Live streaming applications reveal that only channels with very high population of viewers are capable of satisfying the QoE requirements of users. In order to enable bandwidth-constrained devices and channels to also satisfy the QoE requirement, we design and describe a new streaming framework for P2P Live networks termed ACRS. Bandwidth requirement for the ACRS scheme is minimized within through the use of semi-permanent substream distribution groups per bandwidth-constrained channel. The need for fair and accurate assignment of peers to respective substream distribution groups gave rise to Peer Reassignment Algorithm (PRA) and its flowchart diagram. Our future work will first focus on enhancing existing PRA so that at a time, the enhanced PRA can adequately cater for entire k-channel substream distribution groups whose resource indices is below the minimum required.

KEY WORDS: P2P Live networks, ACRS, Substream Distribution Groups, PRA, Enhanced PRA.

I. INTRODUCTION

There are numerous P2P applications in use today each possessing the features of P2P networks as classified in [1]. In existing P2P Live streaming systems such as Joost [2, 3], Zattoo [4], SopCast [5], etc peers viewing the same channel are responsible for redistributing the channel's video stream to their neighbouring peers. For instance, *nth*-peer who intends to view channel X sends collaboration request to other viewers of the same channel. After *nth*-peer had receive an acknowledgement of its request as earlier sent, it begin to receive video chunks accrued to channel X and can only enjoy viewing the said channel after it had continuously saved some chunks in its buffer thereby contributing to longer start-up delay as defined in [1]. Further, *nth*-peer can only distribute the video chunks for channel X. If *nth*-peer switches from channel X to channel Y, it has to send a collaboration request to a group of peers viewing channel Y and wait for subsequent acceptance from the group. Subsequently, these series of actions needed to switch from one channel to another contributes to longer channel switching delays which is part of the parameters defined in [1] that contribute towards achieving the channel's best quality.

Our previous work [6] investigates the collective impact of the independently defined Quality of Service (QoS) parameters in [1] (i.e. start-up delay, channel-switching delay and playback continuity) towards achieving satisfactory state on viewing peers and channels. Results obtained reveal that channels with high population of viewing peers certainly achieve satisfactory state while channels with fewer numbers of viewing peers certainly achieve unsatisfactory state.

However, results obtained in [6] will not in any way enable the future dream of P2P networks to come to fruition, i.e. single user generated videos from existing ubiquitous devices such as iPhones, palmtops, smartphones, PDA's etc be streamed to interested viewers over the Internet. This is due to the low number of interested viewers as well as their aggregate bandwidth. Hence, the need for re-architecturing of existing P2P Live streaming networks in order to also support channels with low viewers and bandwidth to also achieve satisfactory state in the nearest future.

By the nature of P2P systems, peers not only consume resources but also contribute their own resources (i.e. bandwidth) to the network hence, the more the number of peers on a channel, the better the streaming quality. A change in the number of peers viewing a channel would significantly affect the streaming quality of that channel. As such, channels with larger population of viewers would have bandwidth resources more than required to achieve satisfactory state hence, a large portion of bandwidth resources are unutilized. At the same time, channels with low population of viewers

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 6, Issue 8 , August 2019

are suffering much because of inadequate bandwidth resources to achieve channel satisfactory state. These shortcomings of existing P2P Live video streaming systems led to the proposition of a new streaming framework in this paper called Across Channel Resource Sharing (ACRS) scheme.

We present a detailed description of ACRS-based P2P Live streaming framework in section II. We then acknowledge and minimize the bandwidth requirement in ACRS framework when compared to existing P2P Live streaming frameworks in section III. Due to the heterogeneous nature of participating peers, we design an algorithm with its flowchart diagram that fairly assign peers to substream distribution groups according to demand in section IV. Section V concludes the paper.

II. ACRS FRAMEWORK

In order to adequately support “poor channels” to also achieve satisfactory state, the unutilized bandwidth for high uploading peers on “good channels” is properly harnessed by the ACRS scheme. As such, peers with high uploading capacity are allowed to choose a channel of interest (i.e. for viewing and video stream distribution simultaneously) and are additionally assigned to one or more number of channels for distribution purposes only (i.e. based on their available upload bandwidth capacity) hence, a peer distribute a video stream of the channel it is viewing plus additional stream(s) of other channel(s) assigned to it.

In ACRS scheme, a distribution group is designed for all the channels. This is specially designed to enhance the performance of less popular channels. To reduce the bandwidth burden on peers in distribution groups, a channel’s video stream is divided into substreams and a distribution group is assigned for each substream. The server only uploads each substream to one peer in each distribution group. As such, a semi-permanent distribution group is made for each channel. In this case, peer churn will have less effect on unpopular channels since there is a semi-permanent distribution group. A peer is engaged in uploading and viewing channel P (as its channel of interest) and only uploading a substream for channel Q (as its assigned channel). Using the one-to-many relationship between distribution peers and viewers, viewers obtain all substreams from all distribution groups.

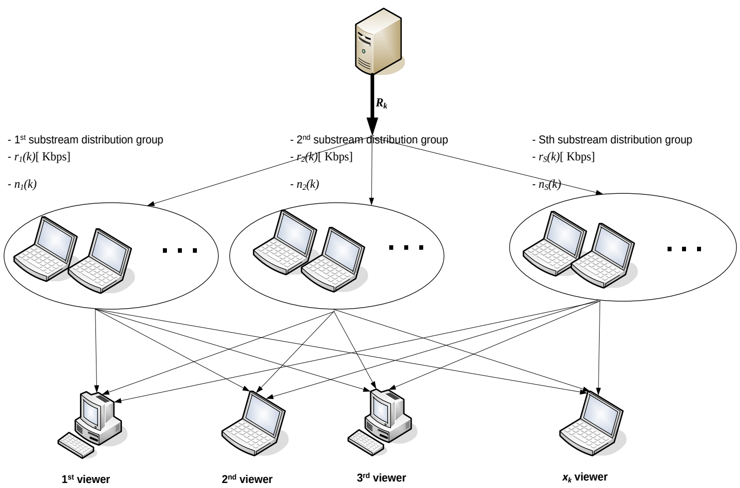

Figure 1: *k*-channel ACRS-based P2P Live Streaming Framework

Figure 1 describes an ACRS-based *k*-channel P2P Live streaming network. It shows how the unpopular *k*-channels’ streaming rate is divided into three (3) equal substreams each of which is streamed to its appropriate substream distribution group. It also presents the fact that *k*-channel viewers must obtain all substreams from each group in order to watch the entire video with good viewing quality.

Furthermore, if a peer is assigned to only one substream distribution group, it obtains the entire substream of that group and redistributes it to other peers in its distribution group and to viewers of its corresponding channel. If a peer is assigned to more than one substream distribution group, it will cater for the redistribution of more than one substream by devoting part of its upload bandwidth to its assigned distribution groups. Since peers are assigned to distribution groups, the assignment must be fairly and accurately done according to demand. That is, the aggregate upload bandwidth for distribution peers devoted to their distribution group should reflect the demand for the substream they redistribute. ACRS scheme also allows various distribution mechanisms within a distribution group.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 6, Issue 8 , August 2019

However, substream distribution within a distribution group may be viewed from two (2) perspectives. One such perspective assumes that peers within a substream distribution group relate with one another via a super-peer architecture; the super-peer being responsible for obtaining the entire substream from the server and distributing it to all members of the same group as shown on figure 2a. Alternatively, the recipient of the entire substream may stream it to a few members of the same group who ultimately distribute same substream to entire members of the same group through the well-known P2P fashion as exemplified in figure 2b. Due to the real-time nature of P2P Live applications and their high demand by viewing peers, we recognise the super-peer architecture for the design of each substream distribution group depicted on figure 2a.

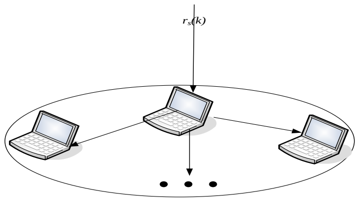

Figure 2a:Super-peer based distribution group

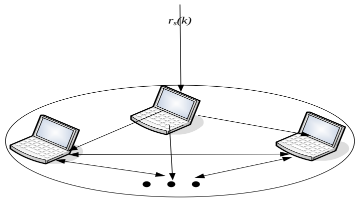

Figure 2b:P2P-fashioned distribution group

III. ACRS BANDWIDTH OVERHEAD

The ACRS framework requires more bandwidth than the original P2P streaming design where a peer distributes only for the channel it is viewing. To minimize the bandwidth overhead, the ACRS scheme propose that a channel is divided into a number of substreams each of which is distributed to viewers of that channel by a group of distribution peers. So a distribution peer is obliged to view and distribute video streams for its channel of interest and to download and upload a substream for a different channel simultaneously.
Let's define the ACRS bandwidth overhead for *k*-channel as the ratio of the aggregate upload bandwidth on entire *k*-channel distribution groups to the required bandwidth to serve *k*-channel viewers. Thus, there is a need to minimize the ACRS bandwidth overhead. To this end, we consider *s*-substream distribution group of the *k*thACRS channel with the following notations:

$r_s(k)$ – substream rate for the *s*-substream distribution group, where $\sum_{s=1}^{S} r_s(k) = R_k$ and $\sum_{k=1}^{K} R_k = R$.

$Đ_s(k)$– group of distribution peers for substream *s*

$n_s(k) = |Đ_k|$, where $\sum_{s=1}^{S} n_s(k) = N_k$

u_i^s – Upload rate of peer *i* devoted to the *s*-substream distribution group

x_k – Number of viewers on channel *k*, where $x_k \in X$

For each distribution group, the server uploads to only a single high-uploading peer in a distribution group whose upload capacity should not be less than $(n_s(k)- 1)\, r_s(k)$ for the peer in question to distribute the substream to other distribution peers in the same group.

Proposition I: In ACRS scheme, the minimum bandwidth overhead for channel *k* with x_k viewers has achievable lower bound of $\frac{1}{x_k R_k}$.

Proof:

To serve x_k viewers with *s*-substream, the following inequality must hold.

$$x_k r_s(k) \le \sum_{i \in Đ_s(k)} u_i^s - (n_s(k)-1) r_s(k) \quad (1)$$

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 6, Issue 8 , August 2019

The k-channel ACRS overhead is given by: $\varepsilon^{(I)} = \dfrac{\sum_{s=1}^{S} n_s(k) r_s(k)}{x_k \sum_{s=1}^{S} r_s(k)} = \dfrac{1}{x_k R_k} \sum_{s=1}^{S} n_s(k) r_s(k)$

(A) Case Study

Let's consider the setting of a P2P system at a particular time t where X = 2000 peers, K = 100 channels and all the channels have equal streaming rate R_k= 500 *Kbps*. The server upload each channel with equal rate v_k = 500 *Kbps*. We assume that the system has equal number of high and low upload peers each with upload rate 1500 *Kbps* and 100 *Kbps* respectively. The channel's popularity follows Zipf distribution with the Zipf parameter z = 1 where Zipf distribution is given by:

$$\rho_k = \frac{1}{k^z \cdot \sum_{i=1}^{K} \frac{1}{i^z}} \qquad (2)$$

Each bandwidth-constrained channel possess substream distribution groups S = 5, substream bandwidth requirement $r_s(k)$ = 100 *Kbps* and number of peers per substream distribution groups $n_s(k)$ = 3.

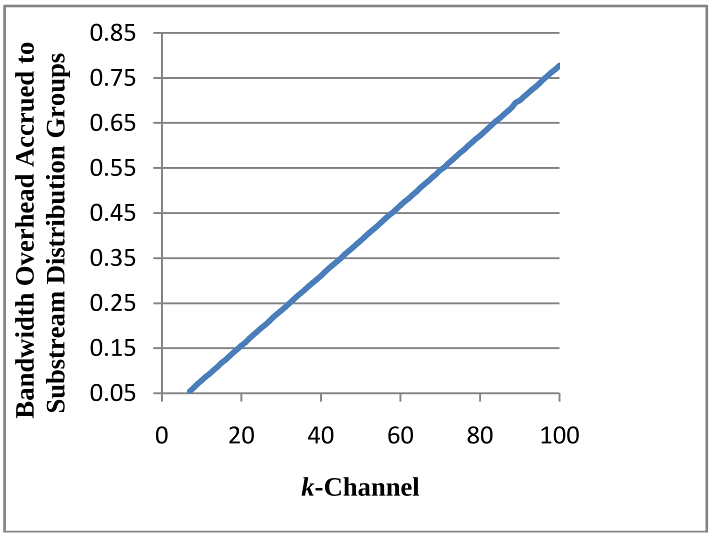

Figure 3: *k*-channel ACRS overhead

Figure 3 presents the bandwidth overhead accrued to Live streaming channels due to the additional bandwidth obtained from their substream distribution groups. Considering a fixed number of distribution groups for all bandwidth-constrained channels, we note that bandwidth overhead is high for less populated channels than it is for more populated channels.

Proposition II: The minimum achievable ACRS overhead for peers' with homogeneous upload rate is feasible when all the substreams are divided equally in terms of the streaming rate. When distribution peers have the same upload rates, then $u_i^s = u$; substream rate $r_s(k)$ as well as the number of distribution peers per substream group is also assumed to be the same. Therefore, $r_s(k)= r$ and $n_s(k) = n$ so that we have:

$$x_k r \leq nu - (n-1)r \qquad (3)$$

$$n \geq \frac{(x_k - 1)r}{u - r}$$

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 6, Issue 8 , August 2019

$$\sum_{k=1}^{K} N_k R_k \geq \sum_{k=1}^{K} \frac{(x_k - 1)R_k^2}{u - R_k}$$

Since $\frac{R_k^2}{u - R_k}$ is a convex function over $R_k \in [0, u]$, through Jensen's inequality [7], we have:

$$\sum_{k=1}^{K} \frac{R_k^2}{u - R_k} \geq K * \frac{\left(\frac{1}{K}\sum_{k=1}^{K} R_k\right)^2}{\left(u - \frac{1}{K}\sum_{k=1}^{K} R_k\right)} = \frac{R^2}{uK - R}$$

Hence, the minimum ACRS overhead for *k*-channel is given by: $\varepsilon^{(II)} = \frac{(x_k - 1)R_k}{x_k(uS - R_k)}$

Proposition III: The minimum achievable ACRS overhead for peers with heterogeneous upload rate is also achieved when all the substreams are divided equally. Because distribution peers have different upload rates, they are assigned to distribution groups so that the average aggregate upload rate in each group is at least equal to the minimum required. Therefore, $u_i^s \approx \bar{u}$ and $r_s(k) = r$ so we have:

$$x_k r \leq n_s(k)\bar{u} - (n_s(k) - 1)r \qquad (4)$$

$$n_s(k) \geq \frac{(x_k - 1)r}{\bar{u} - r}$$

$$\sum_{k=1}^{K} N_k R_k \geq \sum_{k=1}^{K} \frac{(x_k - 1)R_k^2}{\bar{u} - R_k}$$

Since $\frac{R_k^2}{\bar{u} - R_k}$ is a convex function over $R_k \in [0, \bar{u}]$, through Jensen's inequality, we have:

$$\sum_{k=1}^{K} \frac{R_k^2}{\bar{u} - R_k} \geq K * \frac{\left(\frac{1}{K}\sum_{k=1}^{K} R_k\right)^2}{\left(\bar{u} - \frac{1}{K}\sum_{k=1}^{K} R_k\right)} = \frac{R^2}{\bar{u}K - R}$$

Hence, the minimum ACRS overhead for *k*-channel is given by: $\varepsilon^{(III)} = \frac{(x_k - 1)R_k}{x_k(\bar{u}S - R_k)}$

(B) Case Study

Let's consider the setting of a P2P system at a particular time *t* where X = 2000 peers, K = 100 channels and all the channels have equal streaming rate R_k= 500 *Kbps*. The server upload each channel with equal rate v_k = 500 *Kbps*. We assume that the system has equal number of high and low upload peers each with upload rate 1500 *Kbps* and 100 *Kbps* respectively. The channel's popularity follows Zipf distribution with the Zipf parameter z = 1 where Zipf distribution is given in equation (2) above.

Similarly, each bandwidth-constrained channel possess substream distribution groups S = 5, substream bandwidth requirement $r_s(k)$ = 100 *Kbps* and number of peers per substream distribution groups $n_s(k)$ = 3.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 6, Issue 8 , August 2019

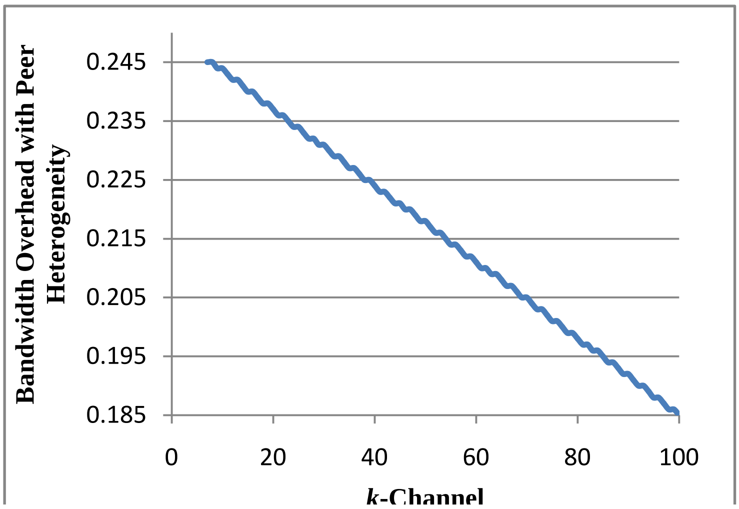

Figure 4: Bandwidth Overhead with respect to Peer Heterogeneity

Figure 4 considers real-life P2P Live networks where peers' possess different upload capacities. With fixed substream distribution groups across bandwidth-constrained channels, figure 4 signify that the incurred overhead is insignificant though least for low populated channels and highest for more populated but bandwidth-constrained channels hence, the future is promising for single user generated Live video-based channels.

IV. ACRS ADAPTIVE PEER ASSIGNMENT

The previous section have acknowledged the existence of bandwidth overhead required by ACRS systems and a derivation of how the overhead can be minimized within (i.e. without recourse to other external bandwidth sources) is shown so that it generates insignificant effect over the entire system. We note that peers naturally possess heterogeneous upload rates and as such, must be assigned to substream distribution groups accordingly. In this section, we propose an algorithm that assign peers to distribution groups so that the assignment result in a balance of the resource indices among substream groups and adapt to variations in channel demand. We further represent the peer reassignment algorithm using a flowchart diagram.

We define the resource index for the *s*-substream distribution group on *k*-channel as $\rho_s(k) = \dfrac{s_s(k) + \sum_i u_i^s}{r_s(k)(x_k + n_s(k))}$

where:

$s_s(k)$ - Server upload bandwidth dedicated to substream *s*

u_i^s - Peer i bandwidth dedicated to distributing substream *s*

$r_s(k)$ - Streaming rate of substream *s*

x_k - Average number of substream *s* viewers

$n_s(k)$ - Number of peers in distribution group *s*

For each ACRS channel, we define the minimum and mean resource index for its substream distribution groups, $\rho_{\min}$ and ρ_{mean} respectively so that $\rho_{mean} \geq \rho_{min}$.

$u_{\min}^s$ - Minimum upload bandwidth required to join substream distribution group *s*

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 6, Issue 8 , August 2019

Considering ACRS *k*-channel, we arrange its substream distribution groups in order of their ascending resource indices (i.e. $\rho_{least},..., \rho_{highest}$). We then assign peers to distribution groups accordingly, ensuring that $u_i^s > u_{\min}^s$. Where some distribution groups still have their $\rho_s(k) \le \rho_{\min}$, we adjust peer assignment by moving peers with lowest upload bandwidth contribution from the group with $\rho_{highest}$ to the group with ρ_{least} to achieve stability within *k*-channel. Below is the peer reassignment algorithm and its flowchart diagram:

Algorithm A: Peer Reassignment Algorithm (PRA)

Step 1. Initialize A = {Distribution groups arranged in order of resource indices increase i.e. $G_1(\rho_1(k)),..., G_s(\rho_s(k)),..., G_S(\rho_S(k))$ since $\rho_1(k) < ... < \rho_s(k) < ... < \rho_S(k)$ };

Step 2. Initialize $B = \{ G_1(\rho_1(k) < \rho_{\min}),..., G_i(\rho_i(k) < \rho_{\min}),..., G_I(\rho_I(k) < \rho_{\min}) \}$;

Step 3. Initialize C = {};

Step 4. while $B \ne \{\}$ **do**

Step 5. Select the distribution group *s* with the highest $\rho_s(k)$ from set *A*;

Step 6. while $\rho_s(k) \ge \rho_{mean}$ **do**

Step 7. Remove the peer with the lowest utilization of upload bandwidth from group *s*, and place the peer into *C*;

Step 8. end

Step 9. Sort the peers $\in C$ in descending order according to their available upload bandwidth;

Step 10. for each peer $j \in C$ **do**

Step 11. From the first to the last, assign *j*-peer to G_i in *B* to increase $\rho_i(k)$;

Step 12. If $\rho_i(k) > \rho_{min}$, remove G_i from *B*;

Step 13. Add G_i to A.

Step 14. end

Step 15. End

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 6, Issue 8 , August 2019

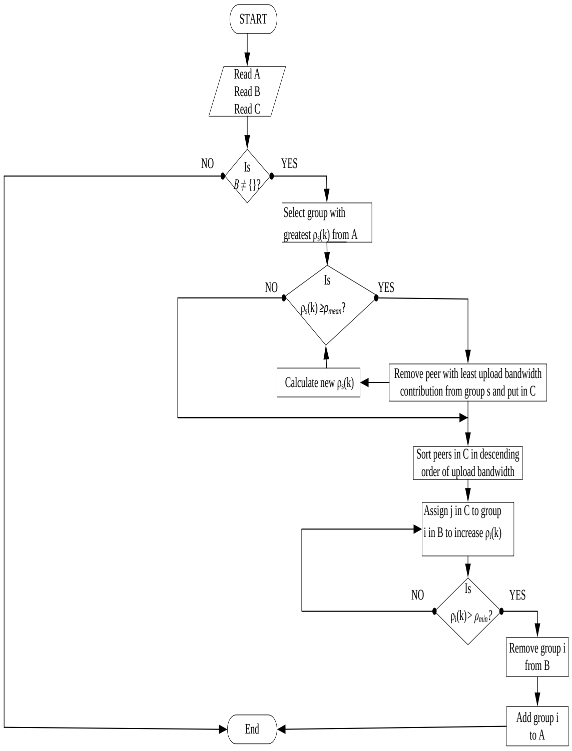

Figure 5: Flowchart Diagram for Peer Reassignment Algorithm

V. CONCLUSION

In this paper, a detailed description of ACRS-based P2P Live streaming framework is presented. The ACRS framework is meant to support future single user generated channels with low popularity to also achieve satisfactory state. The bandwidth requirement for the new P2P Live streaming framework is adequately investigated. We also present an algorithm that fairly assigns peers to the most suitable substream distribution group. We notice that the peer reassignment algorithm caters for a single substream distribution group with the least resource index at a time before improving upon the resource index of the next substream distribution group in that sequence. Our future work will first focus on how to enhance peer reassignment algorithm so that at a time, it caters for any number of substream distribution groups in *k*-channel whose resource indices is below the minimum required by the system. We will mathematically describe the efficiencies of the current and enhanced peer reassignment algorithm in order to establish their similarities and differences.

REFERENCES

1. Aminu, A. and Bashir, A. J. (2015) Taxonomy of P2P Networks and their Associated QoS Parameters. Journal of the Nigerian Association of Mathematical Physics. Volume 31: pp. 95 – 100
2. Agboma, F., Smy, M., and Liotta, A. (2008) QoE Analysis of a Peer-to-Peer Television System. IADIS International Telecommunications, Networks and Systems pp. 114 – 119
3. Krieger, U. R., and Schwebinger, R. (2008) Analysis and Quality Assessment of Peer-to-Peer IPTV Systems. IEEE International Symposium on Consumer Electronics pp. 1 – 4
4. Alhaisoni, M., Liotta, A., and Ghanbari, M. (2009) Performance Analysis and Evaluation of P2PTV Streaming Behaviour. IEEE Computers and Communications pp. 89 – 94
5. Lu, Y., Fallica, B., Kuipers, F. A., Kooij, R. E., and Mieghem, P. V. (2009) Assessing the Quality of Experience of SopCast. International Journal of Internet Protocol Technology, 4: 11 – 23
6. Adamu, A. and Ahmed, J. B. (2017) Analysis of the Probability of Channel Satisfactory State in P2P Live Streaming Systems. Bayero Journal of Pure and Applied Sciences, 10(1): 20 – 27, ISSN 2006 – 6996, http://dx.doi.org/10.4314/bajopas.v10i1.4S
7. Kulkarni, V. G. (2011) Introduction to Modelling and Analysis of Stochastic Systems New York U.S.A., Springer, 313p