Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2009, Article ID 263936, 8 pages
doi:10.1155/2009/263936

Research Article

Theoretical Models for Video on Demand Services on Peer-to-Peer Networks

Giovanni Branca,[1] Thomas Schierl,[2] and Luigi Atzori[1]

[1] *Department of Electrical and Electronic Engineering, University of Cagliari, 09123 Cagliari, Italy*
[2] *Fraunhofer Institute for Telecommunications, Heinrich Hertz Institut, 10587 Berlin, Germany*

Correspondence should be addressed to Giovanni Branca, giovanni.branca@diee.unica.it

Received 3 March 2009; Accepted 20 July 2009

Recommended by Maurizio Murroni

Peer-to-peer networks (P2Ps) are becoming more and more popular in video content delivery services, such as TV broadcast and Video on Demand (VoD), thanks to their scalability feature. Such characteristic allows for higher numbers of simultaneous users at a given server load and bandwidth with respect to alternative solutions. However, great efforts are still required to study and design reliable and QoS-guaranteed solutions. In this paper, within the scenario of P2P-based VoD services, we study the phenomenon of peer churns and propose four models of the peer behaviour to evaluate its impact on the system performance, which are based on the Gilbert-Elliot chain, the fluidic representation of the user behavior, and a queuing analysis of the system. The models are compared by computing the resources the system has to add on top of the P2P network to satisfy all the download requests. Simulations show important relationships between playback buffer length, peer request rate, peer average lifetime, and server upload rate.

1. Introduction

Last years have been characterized by an exponential growth of video traffic on the Internet, which has brought to significant investments in networks and systems aimed at the delivery of real-time high-rate streams. Several traffic analyses tell us that this growth will continue over the next decade, making video streaming applications the ones driving the Internet evolution during the near future [1]. Video on Demand (VoD) is one of these applications, which requires resources able to deliver a video whenever the customer requests it. Realizing a VoD system using the Internet requires architectures tailored to video characteristics. Even if advanced video coding technologies such as Scalable Video Coding (SVC) [2] allow for an efficient representation of the video content towards the transmission over packet networks, VoD service requires guaranteed bandwidths and constrained transmission delays that make it quite difficult to be provided in the traditional Internet architecture.

Typical VoD solutions can be grouped into four categories [3]: centralized, proxy-based, Content Delivery Network (CDN), and Hybrid architectures. In a centralized architecture, the source server manages all clients: it is the simplest and easiest way to implement a VoD system. This solution has the big disadvantages of having a single point of failure, requiring servers with high computational and transmission capabilities that generate unbalanced network loads. Proxy-based architectures are aimed at decreasing the central server load, introducing proxy-servers in strategic points of the network, typically close to the clients. CDNs can be seen as an extension of the proxy-based approach. Accordingly, the video requests are completely handled by edge servers, streaming the content directly to the clients. No requests are forwarded to the central server, as it instead happens in the proxy-based approach whenever the proxy does not have a copy of the requested content. Even if more robust than the centralized solution, major disadvantages limit the diffusion of the proxy-based and CDN approaches. The former *translates* a single point of failure into many points of failure, fractioning central server load to more servers. The latter may ensure high-quality services but it requires big investments for both network and servers deployment and management. Additionally, all these systems have scalability problems; that is, when the number of clients

increases, the only way to satisfy all the incoming requests is to add new servers proportionally.

Hybrid architectures combine the employment of a centralized server with that of a peer-to-peer (P2P) network. Indeed, P2P technologies have been adopted for the deployment of important applications over the Internet, such as file sharing [4] and voice-over-IP (VoIP) [5]. Differently from file sharing, a P2P-VoD network must guarantee the video delivery to the end-user before rigid deadlines. In P2P-VoD, peers support the delivery of the video to other peers using a cache-and-relay strategy making use of their upload bandwidth so as to decrease server load and to avoid network congestions close to the server site. Advantages are a better use of resources and an increased system capacity that allow for the management of higher number of users. P2P networks are also used to realize video broadcast/multicast over the Internet [6]. This technology is attractive because the P2P paradigm has the intrinsic potential to scale with the number of active participants without requiring additional infrastructure deployments: a greater demand generates more resources.

In a peer-to-peer network each peer is free to join and leave the network without notice, bringing to the phenomena of peer churns. These peer dynamics are dangerous for VoD architectures, affecting the integrity and retainability of the service. In the past, many studies have addressed peer churns in file-sharing networks [7, 8], and some others focus on P2P-VoD systems proposing different techniques to avoid service disruption due to peer churns [9–11]. Differently from these works, this paper does not propose any new solution but analyses the user behaviour so as to develop models aimed at evaluating the impact of the peer churns on the system performance. Four models are then proposed. The first two rely on the Gilbert-Elliot model to represent the user connected and disconnected states; the third one is based on a fluidic analysis of the system; the last one makes use of the queuing theory to represent how the video chunk download requests are processed by the system. The models are compared by computing the resource that system has to add on top of the P2P network to satisfy all the download requests. The importance of an accurate modelling of the churns lies on the possibility to analyse important relationships between system parameters, such as playback buffer length, peer request rate, peer average lifetime, and server upload rate, which can then be used to drive the dimensioning and optimization of system resources while assuring user satisfaction.

The paper is organized as follows. Section 2 illustrates a common peer-to-peer Video on Demand scenario, which represents the basis of our analysis. In Section 3, the proposed theoretical models are described and in Section 4 numerical analyses are presented. Section 5 draws final conclusions.

2. The P2P-VoD Scenario

In a typical P2P-VoD scenario a centralized server receives video requests whereas a number of peers download and upload the same content. This is referred to as a single-video approach, and it differs from the multiple video approach because one peer can share only a video, which is the one it is playing back [12]. In case that all requested content cannot be provided by the peers, the server also streams the content accordingly.

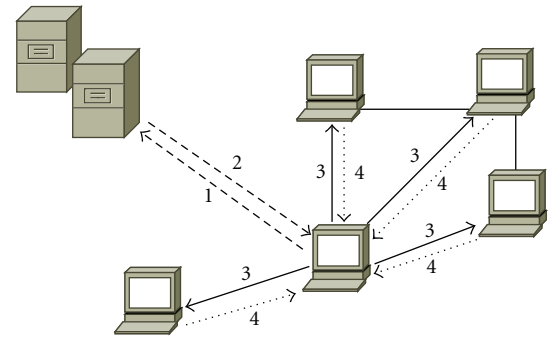

Figure 1: P2P-VOD system: 1: contact server; 2: receive peer list; 3: create application level connections; 4: send data.

Data and control information exchanges can be summarized in few steps. When a new peer joins the system, it contacts the server to know the available video contents. It chooses the video it is interested in and the server sends a list of possible peers that are viewing the chosen content; the peer then tries to create the necessary number of unicast connections with other peers to receive the content and start playing back. When a contacted peer had accepted a connection request, it starts to send useful data. This procedure is illustrated in Figure 1.

Each peer has a playback buffer used to decouple network dynamics from video playing. If a contacted peer does not have the requested data at that moment or it does not reply to the contacting peer, the latter starts creating another connection with the next peer according to the list provided by the server. The server takes charge of distributing a refreshed peer list to all peers whenever necessary, assuming a central role in the coordination of the VoD service.

The most critical problem in a P2P-VoD network is related to the dynamics of peer's participation. In a pure file-sharing network, it is not a serious problem: there are no deadlines to be respected, and it may not be a vital matter if the file download takes more than the expected or desired time. Instead, in the scenario of streaming applications we are considering, peer churns become an important issue which needs to be taken into account to make the system reliable enough to provide an acceptable QoS to the end-user.

The video content is divided in a sequence of video units, named chunks. To avoid playback interruptions, a peer must receive the correct sequence of these chunks before its playback deadline. Not to waste bandwidth, each peer can request only one chunk at time to one peer. We assume that each chunk is of the same transmission length T_{UT} (time to complete the transmission) and of the same playback length T_{UP} (time to finish the playback), both expressed in seconds. Typically T_{UT} is greater than T_{UP}, requiring more than one upstream peer (roughly T_{UT}/T_{UP} peers) per downloading peer on average to have a continuous playback of the video

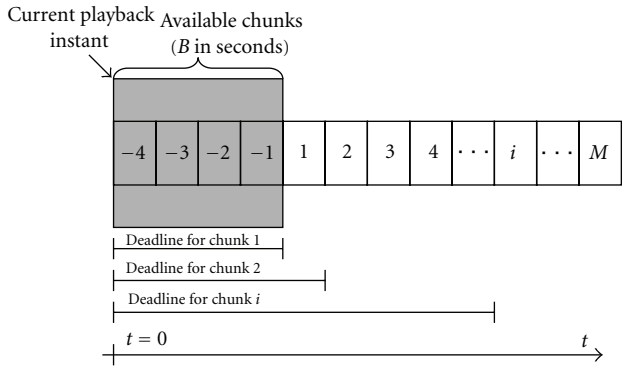

Figure 2: Playback buffer, video chunks, and associated deadlines.

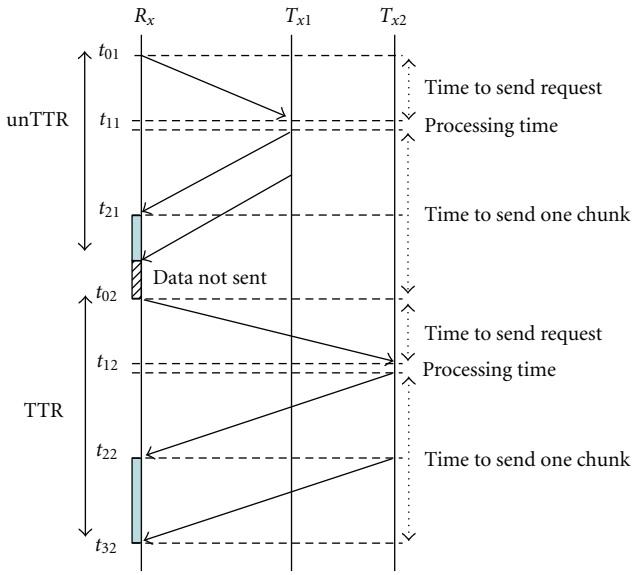

Figure 3: A peer churn event of length unTTR and successive successful request.

without server support. We assume that each peer access line has the same average upload bandwidth U, lower than the video streaming rate R. This is a frequent condition for Internet access in Small-Office Home-Office (SOHO) and domestic users, often characterized by asymmetric access lines.

Figure 2 shows the streaming time-line. The peer has filled its playback buffer and is then starting the playing back. Chunks are enumerated in an increasing progressive way, and the playback buffer B, measured in seconds, is of $4 * T_{UP}$ in length in this example. Chunks from 1 to M are not available yet and are in download phase from other peers at rate U. The deadline for every chunk i is

$$D_i = B + (i-1)T_{UP}. \tag{1}$$

Every time a disconnection occurs, the peer must contact a new available peer. We name the time necessary to complete a correct transmission Time-to-Redirect (TTR), as described in Figure 3.

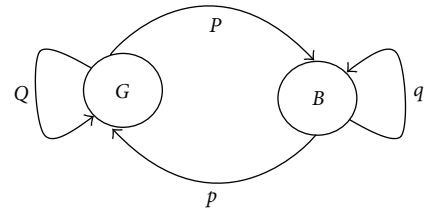

Figure 4: Gilbert-Elliot model.

On the contrary, the unTTR is the time wasted because of one disconnection and it is less or equal than TTR. For simplicity, we consider the worst case taking always unTTR equal to TTR. The TTR depends from many factors, such as nearness of other peers, popularity of video content, and network load.

Due to limited buffer capacity, peers can tolerate up to a maximum number of churns. When the total number of churns is becoming too high for a chunk transmission to a peer, the server takes part in the process by directly sending the chunk. In this scenario, it is interesting to evaluate which is the impact of churns to the whole system. In the following, we describe the proposed four theoretical models to represent peer churns in a P2P-VoD system.

3. Models

In this section we present the proposed models. The first two are based on the Gilbert Elliot (GE) model, the third one relies on Fluidic analysis, and the last is based on the Queueing theory.

3.1. GE Model. In this work, we initially model the peer behaviour using a two-state discrete-time process in which the time axis is measured in terms of TTR intervals. Such a process is then represented with a GE model [13, 14] drawn in Figure 4.

The transition probability P refers to the progress of the peer from the connected-state (good state G) to the disconnected-state (bad state B) during an interval TTR, whereas probability p refers to the inverse process.

Differently, Q and q refer to the probability to remain in the good and bad state, respectively, for an entire interval TTR. In our model, transition probabilities are changed time by time to represent changes in the user behaviour. This probability is taken randomly according to a uniform distribution because peer behaviour is considered stateless and peer participation is supposed very unpredictable. The uniform distribution is left constant for the entire session.

Based on the deadlines described in Section 2, the maximum tolerable number of disconnections is defined as

$$N_{\mathrm{DISC},i} = \frac{D_i}{\mathrm{TTR}}. \tag{2}$$

Each chunk has its own deadline, which has to be met not to interrupt video playback. The probability to satisfy the

deadline condition for a generic chunk i is:

$$\psi_i = \sum_{k=1}^{N_{\mathrm{DISC},i}} P^{k-1} * Q = P_{S,i}. \tag{3}$$

This condition has to be fulfiled for every chunk that a peer is downloading. The probability to fulfil this condition is

$$\psi = \prod_{i=1}^{M} \psi_i. \tag{4}$$

Considering the streaming rate R and the number of peers N into the system, the total bandwidth W_{TOTAL} requested by the whole system is

$$W_{\mathrm{TOTAL}} = R * N. \tag{5}$$

Instead, the peers can provide an upload bandwidth W_{PEER} equal to

$$W_{\mathrm{PEER}} = \psi * N * U. \tag{6}$$

Finally, the bandwidth that the peers are not able to guarantee is the difference between (5) and (6): this is the bandwidth W_{SERVER} requested to the server:

$$W_{\mathrm{SERVER}} = W_{\mathrm{TOTAL}} - W_{\mathrm{PEER}}. \tag{7}$$

3.2. GE Extended Model. The GE model in Section 3.1 is characterized by transition probabilities selected randomly according to a uniform distribution, which is kept constant during the entire video. However, recent studies [15] on user accesses over time, arrival rates, and session lengths have shown that the user behaviour changes during the video playback session. It often happens that the user starts streaming the video and, after a while, he is not satisfied with the content then moves to another video. Accordingly, the probability that a user selects another video is a function of the time, and it decreases as the total amount of played back video increases. Indeed, the probability of streaming interruption is very low after half of the video has been already seen. In particular, it has been proved that the cumulative distribution function of video session lengths is well fitted by an exponential distribution.

Starting from these studies, we propose a GE model extension in which the probability of disconnection P is set according to an exponential distribution: in this way the stay-connect time of each peer is a monotonically increasing function of time, reflecting user trend to stay connected once a significant part of the video has already been watched. Probability of connection p is instead kept constant: its temporal variation's scale is very big if compared with disconnection probability variation, and for this reason it can be considered constant.

3.3. Fluidic Model. Recently, researchers have explored stochastic fluidic analytical models [16, 17] to model traffic in P2P networks. In these models, data transmission is seen like a fluid transferred through nodes, in a similar way to hydraulic models.

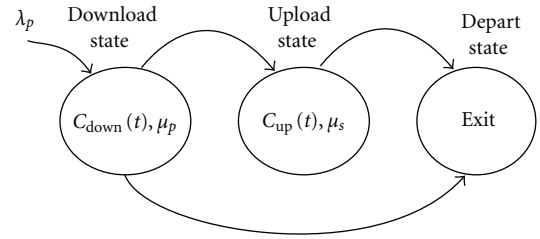

Figure 5: Peer state diagram.

Another study [11] develops a model for P2P-VoD in a broadcast environment. This model can be adapted to P2P-VoD with the hypothesis that peers in upload state can share all video in their memory, not only the first part. Peers can request aid from the server if the P2P network is not able to provide video data, which is the scenario we are considering in this paper. The state diagram of a peer has 3 states: download, upload, and depart, as shown in Figure 5.

When a peer joins the system, it goes in download state and can receive the first part of the video by the P2P network. Therefore, if its playback buffer is full, it goes to the upload state where it can share video parts already downloaded. Finally, a peer can leave the system and moving into the depart state.

The final target is reducing server load in the download state using upload capabilities of peers. From queueing theory point-of-view, the whole system can be approximated as a tandem queueing network with arrival and departure Poisson processes. Given the following:

$$\begin{aligned}
&\lambda_p \quad \text{Arrival rate;}\\
&\mu_p = \frac{1}{\text{First_part_length}} \quad \text{Mean time in Download State;}\\
&\mu_s = \frac{1}{\text{Second_part_length}} \quad \text{Mean time in Upload state;}\\
&\gamma_p \quad \text{Mean Life Time;}\\
&C_{\mathrm{down}}(t) \quad \text{Number of peers in Download state;}\\
&C_{\mathrm{up}}(t) \quad \text{Number of peers in Upload state;}
\end{aligned} \tag{8}$$

it can be developed a simple fluid model to study the system evolution. Peers number in the first state can be calculated considering their exponential distribution, which is proportional to ratio between peer's arrival rate and both mean life time and mean service time (9):

$$C_{\mathrm{down}}(t) = \frac{\lambda_p}{\gamma_p + \mu_p}\left[1 - e^{-(\gamma_p+\mu_p)t}\right]. \tag{9}$$

Instead, peers number variation in upload state is equal to the difference between peers coming from the download

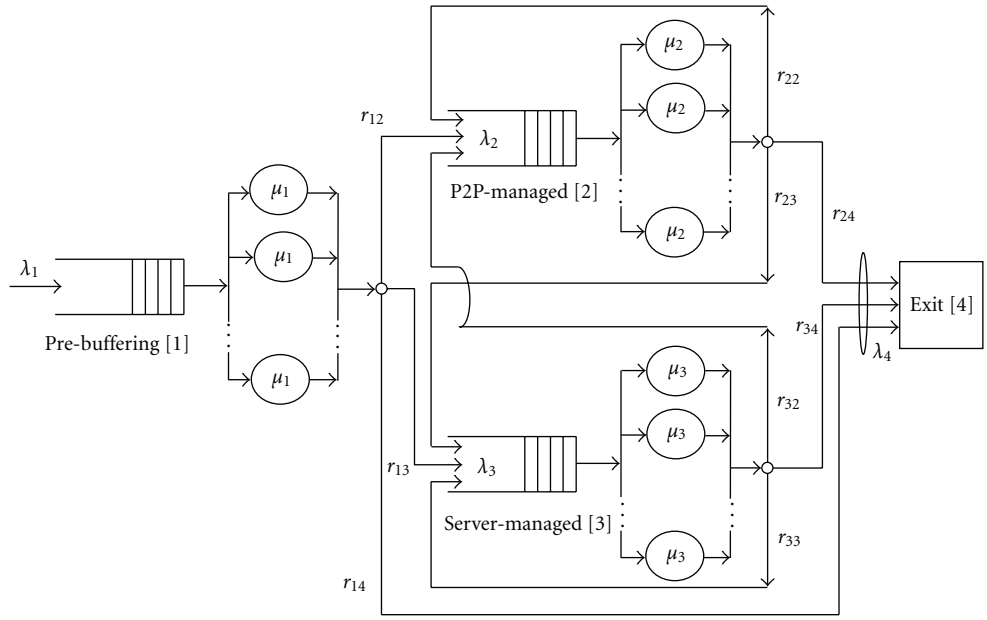

Figure 6: Proposed Queueing model of the P2P-VoD system.

state and the peers going to the exit state:

$$\frac{d}{d(t)}\left[C_{\text{up}}(t)\right] = \mu_p C_{\text{down}}(t) - \min\left(\mu_s, \gamma_p\right) C_{\text{up}}(t). \quad (10)$$

The solution of differential equation (10) is the value of C_{up} as function of the time. The aggregate bandwidth of the P2P network W_{PEER} at time t is equal to $U * C_{\text{up}}(t)$ and bandwidth W_{SERVER} requested by central server is

$$W_{\text{SERVER}} = \left[C_{\text{down}}(t) + C_{\text{up}}(t)\right] * R - C_{\text{up}}(t) * U. \quad (11)$$

3.4. Queueing Model. Queueing theory can be applied to a multiplicity of real problems, especially to transports and telecommunications fields, where each complex system is modeled by a set of queues connected each other. Each individual queue is called *node*, and the state of a queueing network is defined by the simultaneous distribution of customers in each node. In open networks the input rate to a queue i is given by

$$\lambda_i = \lambda_{0i} + \sum_{j=1}^{N} \lambda_j r_{ji}. \quad (12)$$

The term λ_{0i} is the arrival rate of tasks to ith node from outside, and r_{ji} are the routing probabilities that a served task is passed from node i to node j. The term λ_j is the arrival rate of tasks from internal nodes.

A simple queueing network model can be constructed splitting the life cycle of peer in four different phases or states. The first state is a "prebuffering state": peer joins the P2P network and buffers a certain quantity of data before to start video playing. When its buffer is full, it can be routed to the "P2P-managed state," to "Server-managed state" or can leave the system going in "Exit state." Each state is represented by an M/M/∞ queue except for the exit state. The proposed queuing model is shown in Figure 6.

The M/M/∞ queue model is chosen for its analytical tractability. The first queue exactly models the startup delay necessary to fill up the playback buffer. The aim is to collect enough data before starting the video playback to decouple the playback time from the transmission time. The buffer length is fixed, so that the service rate is constant:

$$\mu_1 = \frac{1}{B}. \quad (13)$$

When a peer has filled its buffer, it leaves the first queue and can be routed toward others queues or leave the system. Routing probability depends on the probability to leave the system α and probability to receive data from others peers P_{hit}. For each state, (12) has to be fulfilled as well as the constraint about outgoing routing probabilities for every i:

$$\sum_j r_{ij} = 1. \quad (14)$$

Additionally, the following routing probabilities apply:

$$\begin{aligned} r_{12} &= P_{\text{hit}}, \\ r_{13} &= 1 - \alpha - P_{\text{hit}}, \\ r_{14} &= \alpha. \end{aligned} \quad (15)$$

Exit state could be considered as another queue with service rate unitary: in truth, it is important to calculate only the overall arrival rate to evaluate model dynamics:

$$\begin{aligned} r_{12} &= r_{22} = r_{32}, \\ r_{13} &= r_{23} = r_{33}, \\ r_{14} &= r_{24} = r_{34}. \end{aligned} \quad (16)$$

The mean total number of peers in the system is

$$\overline{X}_{\text{TOT}} = \frac{\lambda_1}{\mu_1} + \frac{\lambda_2}{\mu_2} + \frac{\lambda_3}{\mu_3} = \rho_1 + \rho_2 + \rho_3. \quad (17)$$

Hit probability is calculated dynamically and is proportional to the arrival rate in queue 3 and in exit state:

$$P_{\text{hit}} = 1 - \frac{\lambda_3}{\overline{X}_{\text{TOT}}} - \frac{\lambda_4}{\overline{X}_{\text{TOT}}}. \quad (18)$$

Considering the number of peers ρ_i in each queue i, the bandwidth requested by central server is

$$W_{\text{SERVER}} = (\rho_1 + \rho_2 + \rho_3) * R - \rho_2 * U. \quad (19)$$

Finally, we need to specify the sense of mean service time in queues 2 and 3: every time-step long as mean service time, the next peers' status is set in relationship to the number of peers in queues 2 and 3. If the P2P system contains a sufficient number of peers so that the hit probability is high, this situation influences probability of routing toward P2P-managed state. Otherwise, P_{hit} decreases and it is more probable that a peer will forward to Server-managed state. Notice also that it is not possible to have peers in waiting line because there is always a servant free in an M/M/∞ queue.

4. Simulations

We have performed extensive simulations with different scenarios. The objective of the simulation analysis is to investigate the models behaviour varying the system parameters in order to assess the usefulness of such models in supporting the design and configuration of P2P-VoD architectures. Herein, we present the results when applying the following streaming parameters: transmission of video sequences of 100-minute length at 800 Kbps and an upload rate U of 600 Kbps. We choose these values according to the condition $U < R$, that reflects the most common situation of Internet access lines as explained in Section 2.

The simulations with the GE model has been conducted with a total of 50 peers in the system and changing the stay-connect probability Q every 10 seconds. This probability has been chosen according to a uniform distribution with different ranges, as shown in Table 1. The connection probability p has been kept constant and equal to 0.5 during all simulations.

In the GE extended model, the disconnection probability P follows the exponential distribution (20)

$$P(t) = T * e^{-(t/T)} \quad (20)$$

with the parameter T set so that the complementary stay-connect probability Q has the mean values of Table 1.

To evaluate the effectiveness of these two models, we have computed the requested server download rate at varying disconnection probabilities P. Figure 7 shows the results for the two models when the Time-To-Repair TTR has been set to 300 milliseconds and changing the buffer length from 0.6 to 4.2 seconds.

Table 1: Ranges of the uniform distribution for the probability Q of the GE model.

Range	Values	Mean
#1	0/0.1	0.05
#2	0/0.2	0.1
#3	0/0.3	0.15
#4	0/0.4	0.2
#5	0/0.5	0.25
#6	0/0.6	0.3
#7	0/0.7	0.35
#8	0/0.8	0.4
#9	0/0.9	0.45

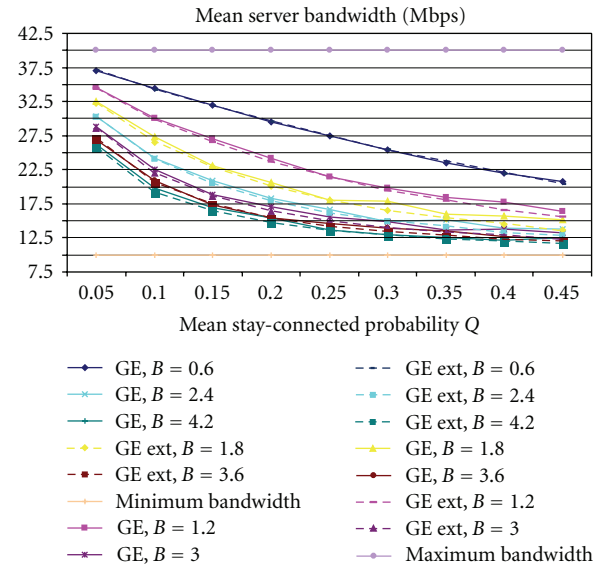

Figure 7: GE models comparison for different values of buffer length.

It can be noted that the two models show similar behaviours, as it was expected since the models are basically the same except the distribution of the connection probability. The shape of the plots shows that increasing the buffer length brings to lower requested bandwidth values. This is due to the fact that the deadline for each chunk is less stringent, allowing for finding an active peer from which successfully download the chunk. The curves are convex so that a higher benefit is obtained by increasing the buffer length at low values of the Q probability. The figure also shows that the total amount of average server bandwidth converges towards 10 Mbps ("minimum server bandwidth" in the figure), which indeed is the difference between the W_{TOTAL} bandwidth of 40 Mbps and the maximum theoretical bandwidth provided by the peers W_{PEER}, which is of 30 Mbps. Overall, this figure is a handy tool that helps the designer in finding the server resources that are required to satisfy the user requests on the basis of the playback buffer length and as far as the operator is able to estimate the peer stay-connected probability. Note that the curve

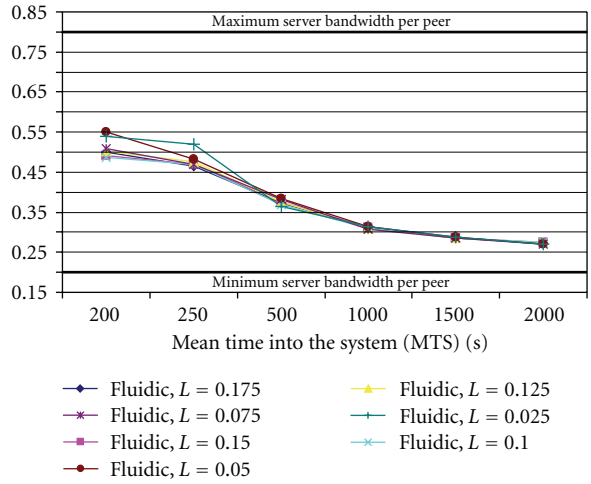

Figure 8: Mean server bandwidth requested by each peer in the Fluidic model with different peer input rates.

"maximum server bandwidth" in the figure represents the amount of server bandwidth that would be necessary without the support of the peer-to-peer network.

In the Fluidic and the Queuing Network models, one of the key parameters is the mean time a peer spends in the system, which is the peer Mean Time in the System (MTS). Whereas for the first model it is directly set by selecting the value of Mean Life Time γ_p, for the Queueing Network model the MTS is indirectly set through the probability to leave the system α, the sampling step Δ, and number of simulation samples N_s according to the following formula:

$$\text{MTS} = \sum_{i=0}^{N_S} \left[(1-\alpha_i)^i \alpha_i \cdot i \cdot \Delta \right]. \tag{21}$$

Equation (21) has been used to find parameters values to achieve the desired MTS. As to the parameter P_{hit}, it has been initialized to 0.9, whereas successive values are dynamically calculated according to the model evolutions.

For the analysis of these models we have computed the mean server download bandwidth requested by each peer, while varying the following parameters: MTS and input arrival rate λ (L in Figure 8). These two parameters affect the number of peers into the system, which then cannot be directly set by us as in the GE models.

Figure 8 shows the requested bandwidth for the Fluidic model. Note that this time the resulting value has been divided by the number of peers in the system, which is different for any combinations of system parameters (see Table 2). In this figure we are also showing the upper and lower bandwidth limits: 800 Kbps is the rate requested to the server when no one peer is able to share video data, whereas 200 Kbps is the difference between the video rate R and the maximum upload rate U, which corresponds to the amount of bandwidth that should be provided by the server when all the active peers are successfully sending video content to another peer.

Table 2: Mean number of peers measured in the Fluidic Model.

Input rate	Mean number of peers					
0.175	34	41	80	146	198	240
0.150	29	36	69	125	170	205
0.125	24	29	58	104	141	171
0.100	19	23	46	83	113	137
0.075	14	18	34	62	85	102
0.050	9	11	23	42	56	68
0.025	4	6	11	21	28	34
MTS	200	250	500	1000	1500	2000

The shape of the plots shows a decreasing bandwidth requested as a function of MTS and implicitly with the increasing number of peers into the system: this behaviour confirms the implicit feature of system scalability of P2P systems. In fact, a bigger number of peers into the system generates more resources (upload bandwidth), reducing the bandwidth requested to server per peer into the system.

5. Conclusions

In this paper we have presented three mathematical models for the evaluation of the peer churn impact on the server resources in P2P-VoD systems. In the first model, the behaviour of each peer is represented by means of the Gilbert-Elliot model, where the two states are associated to the connected and disconnected states. The second and third models use a very different approach with respect to the GE one: a constant number of peers joins the system and the resources requests are related to the effective number of peers inside the system.

The simulations have shown that these models are an effective tool that help the designer in finding the server resources that are required to satisfy the user requests on the basis of the playback buffer length, as far as the operator is able to estimate the peer stay-connected probability. The longer the time each peer spends in the system, the lower the resource required to the server. In fact, an increase in the average stay-connected interval decreases the probability to waste time sending only useless partial chunks from peer to peer, which need to be resent from the beginning by another peer (if available) or by the server.

References

[1] Cisco White Paper, "Approaching the Zettabyte Era," June, 2008, http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481374.html.

[2] H. Schwark, D. Marve, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, 2007.

[3] F. Thouin and M. Coates, "Video-on-demand networks: design approaches and future challenges," *IEEE Network*, vol. 21, no. 2, pp. 42–48, 2007.

[4] BitTorrent, http://www.bittorrent.com/.

[5] Skype, http://www.skype.com/.

[6] J. Liu, S. G. Rao, B. Li, and H. Zhang, "Opportunities and challenges of peer to peer internet video broadcast," *Proceedings of IEEE*, vol. 96, pp. 11–24, 2008.

[7] J. Tian and Y. Dai, "Understanding the dynamic of peer-to-peer systems," in *Proceedings of the International Workshop on Peer-to-Peer Systems (IPTPS '07)*, 2007.

[8] D. Stutzbach and R. Rejaje, "Understanding churn in peer-to-peer networks," in *Proceedings of the 6th ACM SIGCOMM conference on Internet Measurement*, pp. 189–202, September 2006.

[9] Y. Guo, S. Mathur, K. Ramaswamy, Y. Shengchao, and B. Patel, "PONDER: performance aware P2P video-on-demand service," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '07)*, pp. 225–230, Washington, DC, USA, November 2007.

[10] Z. Liu, Y. Shen, S. Panwar, K. W. Ross, and Y. Wang, "Efficient substream encoding and transmission for P2P video on demand," in *Proceedings of the Packet Video*, pp. 143–152, Lausanne, Switzerland, 2007.

[11] K. M. Ho and K. T. Lo, "A simple model for peer-to-peer video-on-demand system in broadcast environment," in *Proceedings of the International Conference on Information Networking (ICOIN '08)*, pp. 1–5, January 2008.

[12] C. Huang, J. Li, and K. W. Ross, "Peer-assisted VoD: making internet video distribution cheap," in *Proceedings of the International Workshop on Peer-to-Peer systems (IPTPS '07)*, Redmond, Wash, USA, 2007.

[13] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Systems Technical Journal*, vol. 39, pp. 1253–1265, 1960.

[14] E. O. Elliot, "Estimates of error rates for codes on burst-noise channels," *Bell Systems Technical Journal*, vol. 42, pp. 1977–1997, 1963.

[15] H. Yu, D. Zheng, and B. Y. Zhao, "Understanding user behavior in large-scale video-on-demand systems," in *Proceedings of EuroSys*, pp. 333–344, Leuven, Belgium, April 2006.

[16] R. Kumar, Y. Liu, and K. Ross, "Stochastic fluid theory for P2P streaming systems," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 919–927, May 2007.

[17] Y. Lu, A. Zhang, H. He, and Z. Deng, "Stochastic fluid model for P2P content distribution networks," in *Proceedings of Autonomous Decentralized Systems (ISADS '05)*, pp. 707–712, April 2005.