



A Bayesian approach for user aware peer-to-peer video streaming systems

Ihsan Ullah^{a,*}, Guillaume Doyen^a, Grégory Bonnet^b, Dominique Gaïti^a

^a ERA/Institut Charles Delaunay – UMR 6279, Université de Technologie de Troyes, 12 rue Marie Curie, 10010 Troyes Cedex, France

^b MAD/UMR 6072 GREYC, University of Caen Lower-Normandy (GREYC), Boulevard du Maréchal Juin BP 5186 – 14032 Caen Cedex, France

ARTICLE INFO

Available online 16 February 2012

Keywords:

P2P live streaming
User behavior
Bayesian networks
Learning

ABSTRACT

Peer-to-Peer (P2P) architectures for live video streaming has attracted a significant attention from both academia and industry. P2P design enables end-hosts to relay streams to each other overcoming the scalability issue of centralized architectures. However, these systems struggle to provide a service of comparable quality to that of traditional television. Since end-hosts are controlled by users, their behavior has a strong impact on the performance of P2P streaming systems, leading to potential service disruption and low streaming quality. Thus, considering the user behavior in these systems could bring significant performance improvements. Toward this end, we propose a Bayesian network that captures all the elements making part of the user behavior or related to it. This network is built from the information found in a cross-analysis of numerous large-scale measurement campaigns, analyzing the user behavior in video streaming systems. We validate our model through intensive simulations showing that our model can learn a user behavior and is able to predict several activities helping thus in optimizing these systems for a better performance. We also propose a method based on traces collection of the same user type that accelerates the learning process of this network. Furthermore, we evaluate the performance of this model through exploring its applications and comparison with non-contextual models.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

P2P live video streaming has become a popular application among the Internet users in recent years. In contrast to IP multicast, it does not require a major change in the current Internet infrastructure. It pushes the load of content delivery to end-hosts turning them into relays that not only receive the content but also transmit it to each other. This feature makes them potentially scalable and deployable with a low cost as compared to centralized Client/Server

(C/S) systems which are expensive and do not scale under a large number of users.

End-hosts in a P2P network, called peers, are controlled by users. Users behave independently and their behavior directly impacts the performance of these systems. For example, the abrupt departure of a user disrupts the stream to its neighbor peers. Similarly, high arrival rates degrade the performance of the whole community. These issues have been indirectly addressed through mesh-based designs and by the use of large buffers. Nevertheless, mesh-based systems require peers to exchange the content availability information with each other and then content is delivered through explicit requests. It incurs an overhead and increases playback delay. Moreover, large buffers increase startup delays. To overcome these problems, the management of network

* Corresponding author.

E-mail addresses: ihsan.ullah@utt.fr (I. Ullah), doyen@utt.fr (G. Doyen), gregory.bonnet@unicaen.fr (G. Bonnet), gaiti@utt.fr (D. Gaïti).

topology with awareness of user behavior can play an important role.

Towards this end, we get insights from user behavior measurements in video streaming systems. We extract from these measurements, the components of the user behavior, their relationships with external elements and network performance parameters. From these insights, we model the user behavior in a P2P live streaming system through a Bayesian network. This network is able to learn a user behavior and allows inferences to predict user activities under given conditions. Estimations produced by this network can be used for making P2P live streaming systems aware of users' behavior and eventually improve their performance. Several application might be considered from the departure anticipation of individual users to the estimation of the global population and performance parameters such as the streaming quality. We evaluate the accuracy of this model through simulations and present its applications for the performance improvement of P2P live video streaming systems.

This work is a revised and extended version of our previous work [1]. Additional proposals cover the enhancement in learning rate which is a significant improvement over the previous work, and applications to the estimation of popularity, arrival and departure rates. Furthermore, extended results of the network evaluation over different behaviors and comparisons with non-contextual models are given.

The remainder of the paper is organized as follows. Section 2 presents works related to user behavior modeling. In Section 3, we perform a synthesis of user behavior measurements for the identification of user behavior metrics and their relationships. It is followed by the presentation of our user behavior model. In Section 4, we present the simulation scenarios and results for the validation of our network. Section 5 illustrates the use of our proposal in challenging applications. Finally, Section 6 gives conclusions and future directions.

2. Related work

User behavior in P2P systems has been addressed in several works. These works analyze and model user behavior in all sorts of P2P applications.

Xu et al. [2] present a content-based partitioning scheme to identify P2P applications. Doulamis et al. [3] propose clustering based on the semantic proximities among P2P users. It is aimed at maximizing the probability of a user to locate the content into its own cluster. Leonard et al. [4] apply principles from classic graph theory to study the partitioning behavior in P2P systems. They show that P2P systems are resilient against peer isolation under many practical conditions. They also propose models for improving this resilience. Feng et al. [5] model user behavior in P2P file sharing systems with the help of two-years logs from a real network. Their work is aimed at generating realistic user behaviors for P2P simulations.

Nonetheless, all the above-mentioned works are not valid for P2P live streaming systems, since these latter operate under different conditions. For example, each

packet in a P2P live streaming system has a playback deadline which does not exist in P2P file sharing systems. Moreover, users actively interact with the content they download, which is not the case in P2P file sharing systems.

Concerning the P2P video streaming context, several works [6–10] analyze user behavior through traces collection and measurement to get insights into it. A few others [11,12,10,13] also model some aspects of the user behavior. We briefly discuss these latter, since they are close to our work.

Through a measurement study, Tang et al. [11] observe that a user's elapsed time in a session is positively correlated with his remaining time in that session. Based on this characterization, they propose that a peer with more elapsed time should be chosen as an upstream. Wang et al. [12] use a similar method for the identification of stable peers. They put those peers in the backbone to reduce the impact of churn. This approach has some limitations: it does not consider any contextual information that impact a user behavior. Moreover, all those peers that have recently joined the system are considered unstable, which is not always the case.

Liu et al. [10] study the user behavior in a P2P live streaming system. They observe that peers stability and their bandwidth contribution ratios are impacted by factors such as the streaming quality. They propose models for predicting the longevity and bandwidth contribution ratios of peers. Nonetheless, there are other parameters such as population and delay, which are also important for a user centric design and that are not considered in their work.

A machine learning approach is proposed in [13]. It enables peers to actively detect the load in the uplink of source peers and alert their clients to replace their source. This approach only considers the upload bandwidth contribution as a metric of the user behavior, while others such as stability of the provider peers are also important for the performance.

In our earlier work [14], we proposed a non-contextual approach for estimating the current session duration of a user from the history of its past sessions. It includes estimators based on Exponential Moving Average (EMA) and Bayes' rule. Moreover, a proactive mechanism of switching to a new upstream peer before the anticipated departure of the current upstream peer was proposed. The limitation of this work consists of its non-contextual modeling which does not take into account the external impacting factors.

One common limitation with the above-mentioned works is their lack of consideration of all user behavior metrics and their influential factors. Moreover, they are intended for specific uses such as estimation of bandwidth or stability. P2P networks highly rely on users and a user-aware design requires the ability of estimating all user activities. Therefore, in [1], we proposed a Bayesian network model that takes into account all the known variables related to user behavior. However, a limit of this model lies in the long learning time required by the network before being able to accurately estimate user behavior metrics. In this paper, we propose improvement

to our previous work to minimize the learning time of the network. In addition, we present its extended evaluation results over six kinds of different behaviors. We also explore its applications to system-level variables such as popularity, arrival and departure rate which are important for understanding the system dynamics.

3. Modeling user activities

User activities in P2P live streaming systems have impacting relationships with the network and contextual conditions. To understand these relationships in detail, we collect and synthesize large-scale measurement studies over user behavior. First, we present an analysis of collected measurements to get insights into user behavior and then we describe our Bayesian network.

3.1. Insights from measurements

An abstract view of the synthesized measurements is shown in Table 1. It is evident from this table that online duration is not studied by most of the measurements. It is mainly due the fact that most of the measurements are intended towards P2P systems and currently deployed P2P systems build a separate overlay for each channel. Therefore, differentiating online and session durations in this type of systems does not make sense. On the other hand, session duration, channel popularity and arrivals/departures have received more attention. Concerning download and upload behavior of peers, they are measured in most of the P2P systems. However, in C/S and telco-managed IPTV systems, the clients do not relay content to other clients and hence they have no upload contribution.

The most interesting insights from these measurement reveal the relationships of user activities with each other,

with the contextual conditions and with network parameters. These relationships are in the form of dependencies and once extracted they are helpful for modeling the user activities. We depict them in Fig. 1. The influence of one element on another is shown through a directed edge. The variables shown in this graph are divided into three categories: (1) user behavior metrics; (2) environmental impacting factors and (3) network performance parameters. User behavior metrics are the components of user behavior. Environmental impacting factors are those variables which are not components of user behavior but come from the environment and have an impact on user behavior metrics. Network performance parameters are part of the network and they have impacting relationships with user behavior metrics. These parameters determine the performance of the system and hence are useful for carrying out decisions. We discuss these relationships in light of the observations presented in measurement studies. We take each dependent variable and explain it with all its impacting ones.

Arrival rate: Time-of-day has an impact on the channel joining rate of users. Users' arrival rate into a channel increases in peak usage time [27]. An increase in arrival rate is observed in the beginning of programs. However, this increase is smoother than the departure. Since programs are scheduled according to time therefore time-of-day has an impact on arrival rate.

Departure rate: Departure rate is impacted by both time-of-day and content type. An increase in departure rate is reported in peak hours particularly towards the end of programs [27,6]. Unlike arrivals, departures happen in batches where users leave the channel simultaneously. Moreover, users switch channels during breaks. Since programs ending and breaks are normally scheduled at fixed times, time-of-day has an impact on the

Table 1
Regression of bankruptcy announcement returns on various control variables.

Ref.	Type	System	Period	Method	Metrics				
					OD	SD	CP	A/D	D/U
[15]			Feb. to Nov. 2008			✓	✓		✓
[16]			Apr. to Dec. 2006	Crawler		✓	✓		
[12]		PPLive	Nov. 2006 (about 28 hours)			✓	✓		✓
[6]			2006 -07	Passive/Crawler		✓	✓	✓	✓
[17]			Jun. 2006	Passive		✓			✓
[8]		PPStream	unknown	Crawler			✓	✓	✓
[18]		Zattoo	Mar. 2008 (2 weeks)			✓	✓		✓
[19]	P2P		Mar. 2005 (4 days)			✓	✓	✓	
[7]		Cool-Streaming	Oct. 2006 (1 day)			✓	✓	✓	✓
[20]			Sep. 2006 (1 day)			✓	✓	✓	✓
[21]		Unknown	2006 (11 hours)				✓		
[22]		GridMedia	Jan. 2006 (4.5 hours)			✓			✓
[10]		UUSee	May to Jun. 2008 (5 days)			✓	✓		
[11]		CCTV	Feb. 2005/Jan. 2006 (2 pop. events)	Logs		✓			
[11]			Oct. to Jan. 2004 -05			✓	✓		
[23]	C/S	Akamai	Oct. to Jan. 2003 -04			✓	✓	✓	
[24]		Unknown	2002 (90 days)			✓	✓	✓	
[25]			Apr. 2008 (6 days)		✓	✓	✓	✓	
[26]	IPTV	Telco-Managed	Jun. 2008 (1 month)			✓	✓		
[27]			May to Oct. 2007			✓	✓	✓	

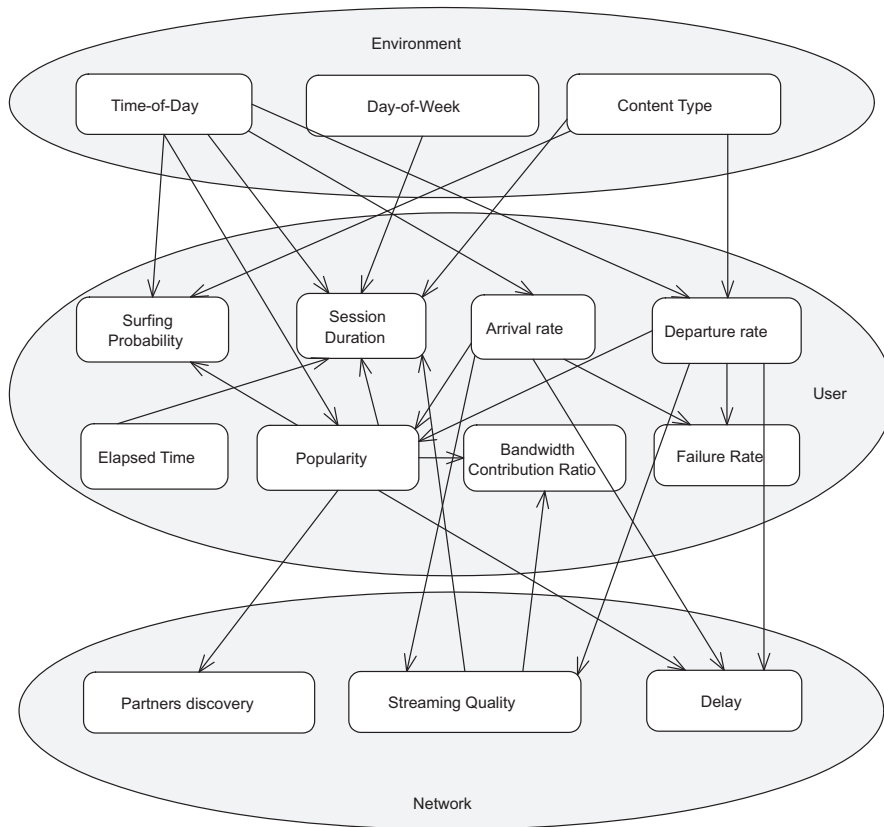


Fig. 1. Components of user behavior and related metrics.

departure rate. The behavior of batch departures is also dependent on the type of the content. For example, Hei et al. [6] observe batch departures in a movie channel but they do not find this behavior in another (unspecified) popular type of channel.

Channel popularity: Channel popularity also called population is impacted by time-of-day, arrival rate and departure rate. According to Qiu et al. [25], popularity is diurnal which means that time has an impact on popularity. Similarly, a higher arrival rate than the departure rate leads to an increase in popularity.

Session duration: Session duration is impacted by the elapsed time in a session, streaming quality, popularity, type of content, time-of-day and day-of-week. A positive correlation is observed between elapsed session duration and remaining amount of session duration of a user [11,27]. Liu et al. [10,28] report a strong correlation between the initial streaming quality and session duration of a user. Session duration is also impacted by popularity. Users stay longer while watching popular programs as compared to unpopular ones [10,6,28]. Similarly, content type impacts the session duration. Shorter session durations are reported for news and music channels while documentaries and kids channels promote longer sessions [27]. Finally, Liu et al. [10] reveal that session duration has a strong correlation with time-of-day but no correlation with day-of-week. However, Veloso et al. [24] observe an impact of day-of-week on the session duration.

Surfing probability: Cha et al. [27] observe that the channel popularity and the type of content impact the surfing (channel browsing) probability of a user. It increases for less popular channels and specific genre like news and music. Studies [27,25] observe diurnal patterns in surfing mode that occur during breaks and at the end of specific TV programs. From this, we can deduce an impact of time-of-day over surfing behavior.

Failure rate: The departure of a user from a channel before the video player becomes ready is called failure. Users' arrivals and departures impact failure rate. It has been found to be strongly correlated with join rate and departure rate [7,29].

Playback delay: Playback delay is the lag between the generation time of a packet and its playback time at the viewing peer. It is increased with an increase in popularity and arrival/departure rates [8].

Streaming quality: Streaming quality is impacted by arrival and departure rates. Departures certainly impact the streaming flow to the dependent peers, but high arrival rates also have the same impact because peers choose randomly their provider peers, which may have joined recently and have not received sufficient video chunks to provide to other nodes. This is also affirmed by [10] that reveals that streaming quality degrades under flash crowds at peak times.

Bandwidth contribution ratio: A user contributes more to the upload bandwidth of the system if he receives a good streaming quality initially. Li et al. [29] observe a

strong correlation between the average bandwidth contribution ratios and the instantaneous popularity.

Partners discovery: An increase in popularity makes the partners finding job easier because of the availability of more peers in the system. Peers face difficulty in finding partner peers while watching less popular channels [19,16].

3.2. A user behavior model

From the synthesis given above, we understand that user behavior has several components and each of them has a relationship with other components, contextual environment and/or network parameters. The goal of our model is to be able to estimate the value of any variable of interest from other variables.

Bayesian networks suit well in this kind of situation. First, they are able to model such a domain and allow multiway inferences in order to estimate different variables of interest through the same model. Second, they are expressive for modeling real-world problems. Finally, they model well the relationships of causes and effects which is the case here.

3.2.1. Bayesian networks

A Bayesian network (BN) is a Directed Acyclic Graph (DAG) of vertices and edges. Vertices, called nodes, represent random variables and directed edges show informational or causal dependencies among variables

[30]. A Bayesian network models the quantitative strength of a relation between two variables and allows to update the probabilistic belief about this relation as new observations become available. A node with a directed edge towards another node is called the parent while the other node is called the child node. Bayesian networks simplify the joint probability distribution of variables through requiring only the set of parent variables. A node is conditionally independent from all other nodes given its parents. The joint probability distribution P of variables $X_1 \dots X_n$ in a Bayesian network can be written in the form shown in (1), where

$$P(X_1, \dots, X_n) = \prod_{i=1}^n p(X_i | \text{parents}(X_i)) \quad (1)$$

The network structure can be constructed by an expert or learned from data. Similarly, the quantitative strength of relationships among variables, called parameters, can be assigned manually or learned from data. In our work, we construct the network manually and let it learn its parameters from the data.

3.2.2. Proposed model

The Bayesian network we propose is depicted in Fig. 2. The structure of our Bayesian network is a direct derivation from the causal graph shown in Fig. 1. The minor adjustments are as follows. We omit day-of-week from this network because it was studied only in two measurements with contradicting observations. We do not assign

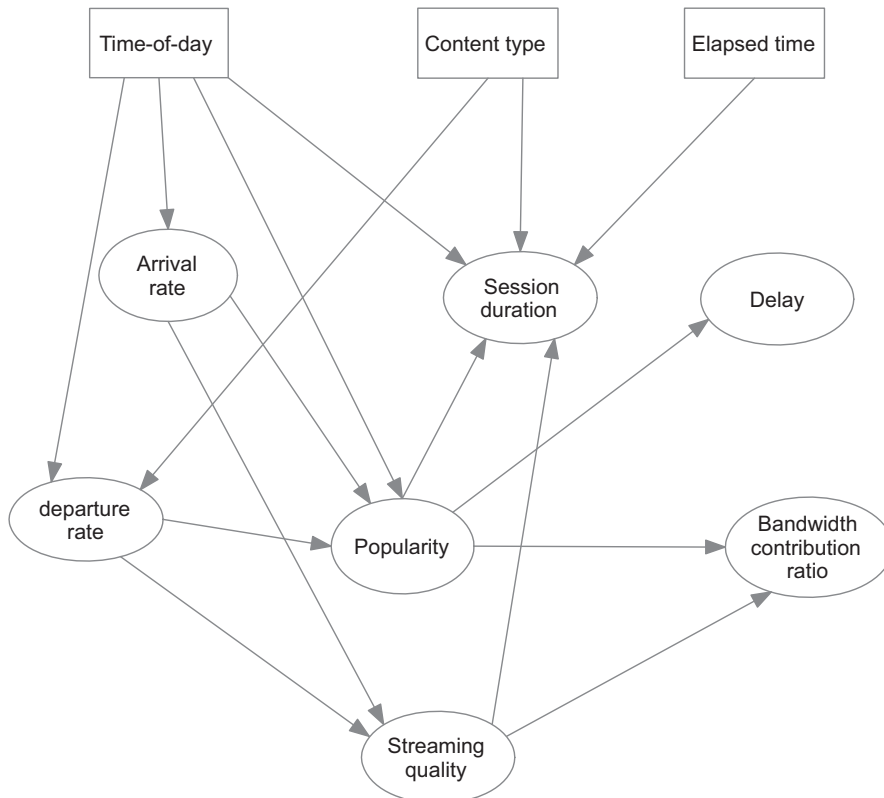


Fig. 2. The proposed Bayesian network for user behavior.

any node to surfing probability and failure rate since both surfing and failure result in short sessions, which can be represented by the session duration variable. Similarly, partner discovery has been excluded since it is relevant only in mesh-based systems and it can be easily added if required.

3.2.3. Nodes' description

Our Bayesian network contains both discrete and continuous variables. We represent discrete variables through rectangles and continuous ones through ellipses. A discrete variable has a state for each possible outcome. We determine the number of these states through the help of measurement studies. Cha et al. [27] analyze several content types and from their observations we group them into three types namely, reality, fiction and sport. Elapsed time of a user can help in deciding whether a user is in surfing or viewing mode. We deduce the number of state for elapsed time from [27], which states that the channel holding time in surfing mode can last up to 2 min. Therefore, we choose binary states for elapsed time: less than 2 min (an unstable peer in surfing mode) and more than 2 min (a stable peer in viewing mode). Since time of the day is discretized, its number of states depends upon the discretization interval. To choose the most appropriate value for the discretization interval, we perform simulations and illustrate them in Section 4.2.2. All the remaining nodes are continuous and they are Gaussian distributed.

4. Validation

In order to validate our model, we perform numerous simulations. Such simulations require the injection of realistic user behavior traces based on users' logs or synthetic models to precisely evaluate how our Bayesian network behaves. However, if the measurement studies presented in Section 3 provide global models, they face difficulties in establishing the link between all traces and individual users due to dynamic IP addresses and peer identifiers, presence of network address translations (NAT) and privacy rules. Moreover, the individual user behavior does not follow the global behavior and a global model is not suitable for dealing with users having different interests and habits. This is why, given the lack of individual traces and accurate individual models, we proposed a new model based on a semi-Markovian process that precisely abstracts the behavior of a user of a P2P video streaming service. Before illustrating the simulations, we briefly discuss this model. For further details about this model please refer to [31].

4.1. Workload model

Inspired from the qualitative description of fictional characters called personas in [32], we propose a formal semi-Markovian model that precisely emulates the behavior of P2P live video streaming users. We assume six personas namely, Johnatan, Emma, Stephan, Anna, Peter and Ellen. We shortly denote them by $\{J, E, S, A, P, L\}$ respectively. We modelize each user with a non-homogeneous

semi-Markovian process. It means that the state of a user (online or offline for instance) at a given time not only depends on its state at the previous time as in any Markovian process but also on the time it spends in this state and on the global time of the process. Such kind of process fits well with the video streaming context because previous studies [10,11] show that the behavior of a user vary with respect to the time of the day and the time spent online. For example, watching longer or more often in the evening than in the morning and the impact of elapsed time on the remaining one.

4.1.1. The abstract individual model

We make the following assumptions:

1. We only consider mono-channel video streaming applications. Consequently, we define two states $\{X_1, X_2\}$ where the semantics are respectively the user's online presence and its offline presence;
2. We assume a cyclic behavior on each day. Consequently, we consider the process global time $t \in \mathbb{N}^+$ as a day discretized in one-minute intervals $\{t_1 \dots t_{1440}\}$;
3. In order to be consistent with [33–36] and as log-normal laws are commonly used to model slow fading phenomenon, we consider that the transition from state X_1 to state X_2 is controlled by a log-normal law. As an individual user watches longer in certain time-of-the day, the $\mu_t^i(d)$ and $\sigma_t^i(d)$ depend on the persona i , on the kind of content d , on the global time of the process t and on t_{X_1} the time spent in the state X_1 ;
4. Since Poisson laws are widely used to model arrival processes, we consider that the transition from state X_2 to state X_1 is controlled by a Poisson law which is consistent with [33,35]. As an individual user has habits about its watching time, the parameter λ_t^i depends on the persona i and on the global time of the process t .

The transition probability from state X_1 to state X_2 is the probability to be disconnected after a given time t_{X_1} spent in the state X_1 :

$$P(X(t) = X_2 | X_1, t_{X_1}) = \int_0^{t_{X_1}} \frac{e^{-(\log x - \mu_t^i(d))^2 / 2(\sigma_t^i(d))^2}}{x \cdot \sigma_t^i(d) \cdot \sqrt{2\pi}} dx$$

where $\mu_t^i(d)$ and $\sigma_t^i(d)$ are the parameters of the log-normal law with respect to the persona i , the global time t and the content d . The transition probability from state X_2 to X_1 is the probability of being connected at a given time-of-the-day. We model this phenomenon with a Poisson law and, as we are interested in the arrival of a single user, $k=1$. Consequently,

$$P(X(t) = X_1 | X_2) = e^{-\lambda_t^i} \lambda_t^i$$

where λ_t^i is the parameter of the law with respect to the persona i and the global time t . In order to instantiate the parameters of the model, we base them on the personas defined by [32]. In the sequel, we denote these personas with $\{J, E, S, A, P, L\}$, namely Johnatan, Emma, Stephan, Anna,

Peter and Ellen. All the parameters and their values are summarized in Table 2.

4.1.2. Arrival model

In conformity with measurements [37,34,35,38], we define five time periods in a day, namely morning, noon, afternoon, evening and night. For each persona and each time period, we set the λ_t^i parameters of our model for a one-minute time step (see the second part of Table 2). We choose those values such that it reflects the watching habits of each persona. For instance, Johnatan watches more often in the evening than in the morning. Moreover, the sum of these values is lesser than 1, meaning that a given user does not join the network every day. Finally, the sum of the λ_t^i approximate the global evolution of a population.

4.1.3. Session duration model

Each persona watches television each day for an average time as given in Table 2. According to [32], J and A are very regular, E and L are regular, P is irregular and S is very irregular. This regularity is given by the variance $\sigma^2 = \bar{x}/k$ where $\bar{x}$ is the total time session per day ($k=1$ for very irregular to 4 for very regular). The third part of Table 2 gives the mean and the variance in minutes of the total online presence for each persona i as well as the μ_i and σ_i parameters of their log-normal laws.

As [32] solely give a watching duration per day for each persona, we need to define $\mu_t^i(d)$ and $\sigma_t^i(d)$ from μ_i and σ_i . Furthermore, the session time is correlated with the interest of the watcher in the content. To this end, we assume that the session time at a given time period (local session time) is a sharing of this watching duration per day multiplied by an interest coefficient defined from both the watching time habits and the socioprofessional group of each persona. The fourth part of Table 2 gives this sharing for each persona according to the time period. The values of these parameters are arbitrary defined but reflect the preferences of the personas. The fifth part of Table 2 gives the interest coefficient θ_d^i for each persona i with respect to each content type d . We choose the values of the coefficients such that their mean equals 1 which represents the fact that, in average, the expected session duration follows the previous distribution. Consequently, we defined $\mu_t^i(d)$ to $\sigma_t^i(d)$ as follows:

$$\mu_t^i(d) = \mu_i \cdot M_t^i \cdot \theta_d^i$$

$$\sigma_t^i(d) = \sigma_i \cdot M_t^i \cdot \theta_d^i$$

4.1.4. Population model

As an individual model, a non-homogeneous semi-Markovian process is instantiated for each user in the

Table 2
Parameters of the model.

Parameter	Persona					
	Johnatan (J)	Emma (E)	Stephan (S)	Anna (A)	Peter (P)	Ellen (L)
<i>Features defined by [32]</i>						
Age	17	25	33	46	58	69
Interests	Sports and serials	No matters	News and sports	Serials	News and reports	Talk-shows and reports
Watching time per day	2–3 h	1.5 h	1.5 h (high Δ)	1.5 h	1.5 h	2 h
Watching time habits	Evening, night	Almost at night	Noon, after work	Afternoon, evening	Noon, evening	No habits
Socioprofessional	Student	Store clerk	Executive	Housewife	Full professor	Retired
<i>λ_t^i values on one-minute intervals with respect to persona i and global time t</i>						
From 4 a.m. to 9:59 a.m.	0.00125	0.00125	0.00125	0.00375	0.00125	0.00375
From 10 a.m. to 3:59 p.m.	0.00375	0.00375	0.0075	0.005	0.005	0.005
From 4 p.m. to 6:59 p.m.	0.00375	0.00125	0.00125	0.00625	0.00375	0.00375
From 7 p.m. to 10:59 p.m.	0.0075	0.005	0.00375	0.00625	0.0075	0.00375
From 11 p.m. to 3:59 a.m.	0.00375	0.00625	0.00375	0.00125	0.00125	0.00125
<i>μ_i and σ_i values on a full day with respect to persona i</i>						
Watching duration per day	150	90	90	90	90	120
Variance	37.5	30	90	22.5	45	40
μ_i	5.0098	4.498	4.4943	4.4984	4.497	4.7861
σ_i	0.0408	0.0608	0.1051	0.0527	0.0744	0.0527
<i>Sharing M_t^i of the total online time with respect to persona i and global time t</i>						
From 4 a.m. to 9:59 a.m.	0.05	0.05	0.05	0.05	0.05	0.25
From 10 a.m. to 3:59 p.m.	0.1	0.05	0.2	0.2	0.3	0.25
From 4 p.m. to 6:59 p.m.	0.1	0.05	0.1	0.35	0.2	0.15
From 7 p.m. to 10:59 p.m.	0.4	0.2	0.4	0.2	0.35	0.25
From 11 p.m. to 3:59 a.m.	0.35	0.65	0.25	0.2	0.1	0.1
<i>Values of the interest coefficients θ_d^i for persona i and content type d</i>						
Fiction	1	1	0.3	2.4	0.3	1
Reality	0.5	1	1.7	0.3	2.4	1.5
Sports	1.5	1	1	0.3	0.3	0.5
<i>Proportion of each persona in the network</i>						
	0.182	0.168	0.177	0.186	0.174	0.113

network. Consequently, we propose to set the proportion of the personas according to the data given by the French National Institute of Statistics and Economic Studies (INSEE). From these data, the proportion of a given persona type is given in the sixth part of Table 2.

4.1.5. Model validation

In order to use such individual models, we need to validate that their global behavior fits with those established in the literature. We set 10,000 users with a persona according to our population model given in Table 2. We simulate three days with a realistic mixed content each day and we compute the average behavior.

Fig. 3 represents the frequency of the session durations of the population. It presents a multimodal distribution where each mode highlights a different behavior with respect to a given content and a given interest coefficient. The distribution shows a high number of short sessions and a long tail of longer session which is consistent with the log-normal model of [39].

Fig. 4 compares the cumulative distribution function (CDF) of the session duration of our model to the global models proposed by [35,36,39]. We can notice that our model fits well with the global model of [35] and have the same shape than the others. Indeed, Jensen–Shannon divergence and the symmetric Kullback–Leibler divergence with [35] is only 0.091486 and 0.48833 respectively. However, [36,39] present more very short sessions. An explanation is that the authors focus on reality shows and soccer where the users watch very short scenes (cues or goal kicks) whereas we consider a realistic mixed content channel.

Fig. 5 represents the population throughout the day. It shows that our model is consistent with the global model presented in [10,16,39]: there is a low population in the morning, a first peak at noon, a decrease in the afternoon, an apex in the evening and a low population at night. The sudden changes in the population represent flash crowds, a common phenomenon in P2P streaming systems. Thereby, our workload model can be used to validate our Bayesian network.

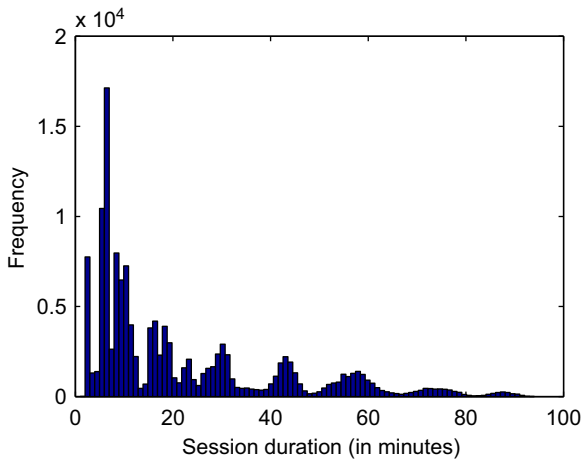


Fig. 3. Session durations' frequency.

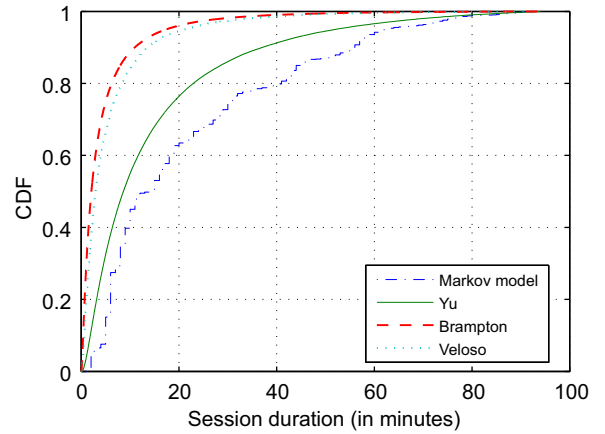


Fig. 4. CDF of session durations.

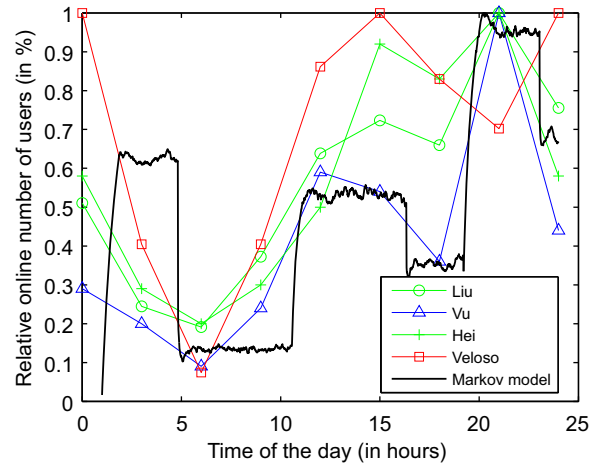


Fig. 5. Population.

4.2. Simulations results

We use Matlab and Bayes Net Toolbox¹ extension for simulations. In this part, we illustrate the simulation scenarios, determination of the number of states for time-of-day and the network accuracy in two deployment types.

4.2.1. Simulation scenarios

The simulation scenarios represent two network deployments. The first one, termed as a global scenario, assumes a central entity for the deployed P2P system. The central entity will have all the required information needed for the network such as time, type of content, arrival departure rates and popularity. The network thus learns a global behavior of users.

The second scenario is called the local one and considers an individual deployment of a Bayesian network in each peer. In the local scenario, although the local information is available, the global parameters that are arrival/departures and popularity need to be provided to

¹ <http://code.google.com/p/bnt/>.

Table 3
Simulation parameters.

Parameter	Value
Total population	1000 users
Simulation duration	40 days (global scenario) 200 days (local scenario)
Content type	Reality, fiction and sport
Content duration	2 h
Inference algorithm	Conditional Gaussian
Learning algorithm	Maximum likelihood learning
Time-of-day discretization interval	1 h (see Section 4.2.2)

the network. Since these parameters are only required soon after joining, it can be provided through a centralized server, which already exist to support other functions such as tracking peers. Moreover, decentralized aggregation protocols [40–42] based on a tree structure or a gossiping approach can also be used to collect this information.

Simulation parameters for these two scenarios are given in Table 3. We simulate a total of 1000 individual users. Users' traces spanning over a 40-days period are used in the global scenario while those over 200-days are used in the local one. The content type is changed after each 2 h duration. We use the Conditional Gaussian algorithm for inference and Maximum likelihood learning algorithm to train the network. The discretization interval for session duration is 1 h. First we present how we choose this interval and then we show the results for the two scenarios.

4.2.2. Time discretization

As we discussed in Section 3.2.3, the number of states for the discrete variable time-of-day and its discretization interval cannot be deduced directly. Choosing a smaller interval enables the network to encode time with finer granularity, hence increasing its accuracy. However, on the other hand, it increases the number of states of time-of-day leading to a larger size of its own CPT and all its dependent variables. Consequently, the required learning time increases, since the network should get data for all combinations of these states in order to be trained. In this situation a tradeoff between accuracy and learning time is required. To find this tradeoff, we perform simulations with different time intervals.

We use the same data set in all simulations. To test different intervals, we ask the network to forecast the session duration of each user when he/she is online. Since session duration has the maximum number of relationships with other variables in the network, its estimation is the most complex one. In the beginning the network has random parameters and it does not have any real knowledge of the behavior it is associated with. Users from different personas are let to join the system. On the arrival of a user, the network is asked to predict the session duration of that user, given all other variables such as time-of-day, content type and population. After the departure of the same user, the observation with real session duration is provided to the network for learning. In this way, the network improves its knowledge as the

time passes on. At the end, actual session durations are compared with the estimated ones. From this comparison, the overall Root Mean Square Error (RMSE) and the learning time are computed. The learning time is the time a network takes after which the variation in RMSE over a ten days period does not exceed a given threshold. We have chosen 0.95 min as a threshold value, which is a small variation in our case. The accuracy of the network is analyzed from the overall RMSE of only those estimations which are produced by the network after the end of the learning period.

The obtained results indicate that our network performs differently in the two scenarios. Since the network deals with datasets of all users in the global scenario, it gets a large amount of data to learn the behavior. It becomes hard to determine the duration in which the network learns. For instance, the RMSE with different intervals from day 2 to 40 only varies between 17.33 and 17.69. These results make the discretization interval invariant regarding the global network performance, since the error is large all the time and there is no notable difference in the learning time over different intervals.

By contrast, results from the local scenario help in choosing the discretization interval. Time-of-day discretization shows a significant impact on the estimation accuracy and learning time in this scenario. We depict the learning time versus different discretization periods in Fig. 6a. Similarly, we show the RMSE after learning of the same networks in Fig. 6b. It is evident from these two figures that increasing the discretization interval decreases the required learning time, while increasing the error. The best tradeoff value we choose is 60 min interval, requiring 24 states of time-of-day. The average learning time for this interval is about 40 days with an average RMSE of about 7.58 min for all personas. From here onward, we use this discretized value in all simulations.

4.2.3. Global case

Our network contains two types of variables. First, peer-level variables that concerns an individual user or peer such as session duration and bandwidth contribution ratio. Second, system-level variables such as arrival/departures rates and popularity. Here, we only evaluate the performance of our network over the estimation of session duration.

Actual session durations versus the estimated ones are depicted in Fig. 7 over a period of 10 days. It is clear from the plot that our Bayesian network over- and under-estimates about the same number of sessions between 0 and 40 min interval. Moreover, estimated sessions uniformly occupy the $[0-40] \times [0-40]$ square indicating a poor accuracy.

On the other hand, the network under-estimates sessions having lengths between 40 and 100 min. Although under-estimating a session duration does not prevent from anticipating the user departure, it induces a strong overhead for any networking mechanism that rely on it. Such a result is due to the fact that the network learns an average behavior from a global trace, which does not help in distinguishing the elements of each user, resulting thus

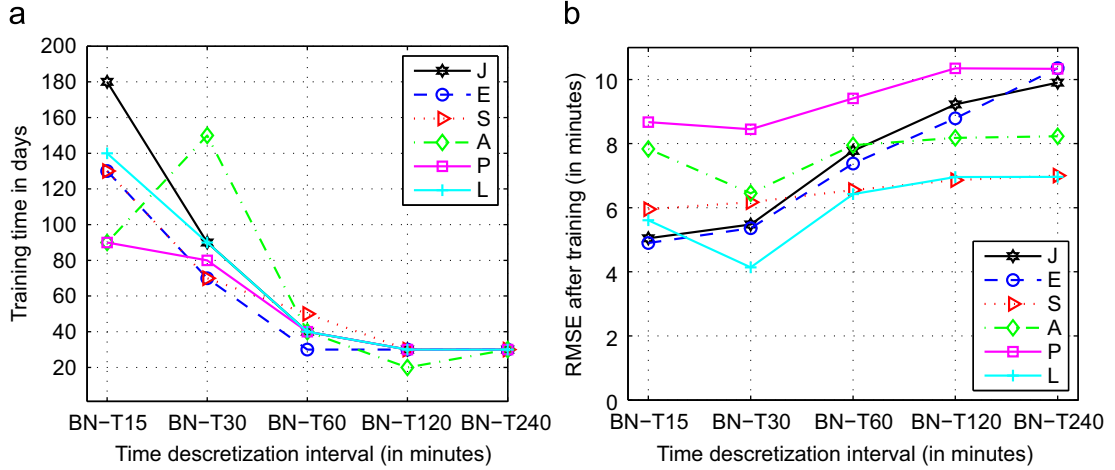


Fig. 6. (a) Learning time. (b) Error. (The number after BN-T indicates the discretization interval in minutes.)

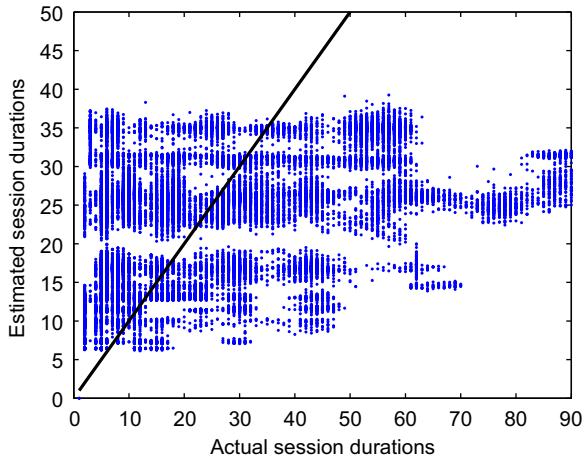


Fig. 7. Actual sessions versus estimated ones in global case.

in shorter or longer sessions for a user of a particular persona type.

4.2.4. Local case

In this scenario too we evaluate our Bayesian network over the estimation of session duration. We apply the traces of individual users of all the six personas to the network and measure its accuracy. We depict the evolution of error and learning in Fig. 8. Here, the RMSE is averaged for all users of each persona over 10 days period. The network shows a maximum error for all personas in the beginning, which is reduced as the time passes on. Understandably, the network learns from observations and minimizes the error. To avoid the redundancy, from here onward, we show results only for persona *L* in the local case, since the network shows a very similar behavior for all personas.

We also show a scatter plot of some representative actual sessions versus estimated ones belonging to 10 users of persona *L* from the 41st day onward in Fig. 9. This plot shows a good precision as compared to the global

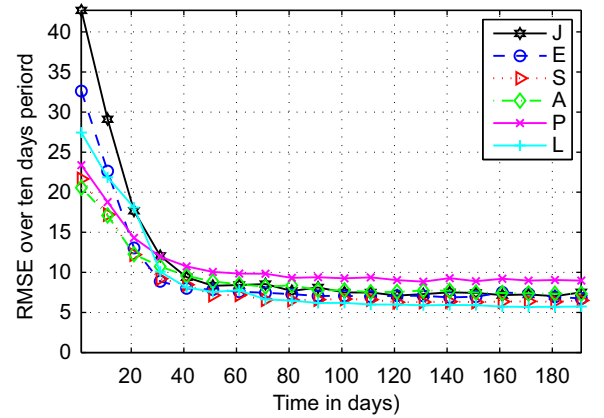


Fig. 8. RMSE averaged over a 10 days period.

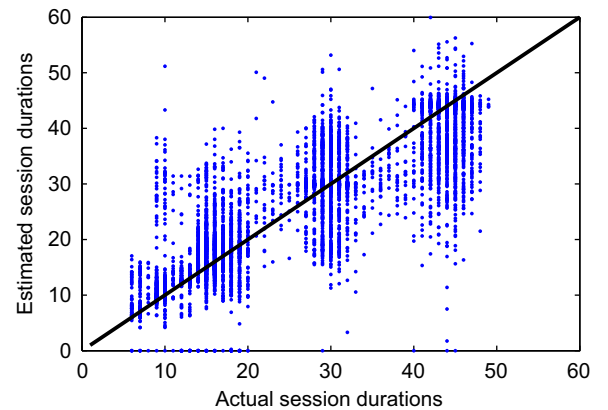


Fig. 9. Actual sessions versus estimated ones.

network, with a slight tendency to over-estimate short sessions and under-estimate long ones. All these results indicate that our network is more accurate in the local

case and its performance is consistent over different kinds of behavior.

4.3. Discussion

It is clear from the above-given results that our Bayesian network does not perform well in the global scenario when the network learns from mixed behaviors and it is asked to predict a peer-level variable. Moreover, for every peer to query a central entity for the estimation of a variable leads to scalability issues. However, this kind of a deployment can be useful for resource allocation at the provider side through estimation of system-level variables such as arrival/departure rates and popularity. We explore the estimation of these variables latter in this paper.

On the other hand, our network shows a good accuracy over the estimation of peer-level variables such as session duration in the local deployment. Nevertheless, the requirement of 40 days learning time emerges as a major limit in the local case, since the network cannot estimate accurately

during this initial period. Therefore, next we propose to reduce this learning period to an acceptable level.

4.4. Increasing learning rate

To enhance the learning rate, we use a combination of a profiler and trace collection from different users of the same persona type. We assume that a profiler initially asks the user about his preferences and then classifies that user into a persona class. To learn the behavior, the Bayesian network of this user then uses the traces of all other users of the same persona class. This mechanism increases the number of observations and thus the network learns quickly. However, the accuracy of the profiler also becomes important. Profiler modeling is out of the scope of this paper, therefore we assume six levels of accuracy for the profiler. Each level corresponds to the amount of percent error that results from the wrong classification of a user. We simulate 50 days for all users of persona L .

We show the RMSE taken over 3-days period in Fig. 10. One can notice that the error is minimized far quicker than individual traces. The learning time is determined from this error evolution. It is the time after which the variation in error does not exceed a certain threshold. To be consistent with the previous method, we set the threshold value to 0.95 min.

Fig. 11a depicts the learning time for each considered profiler. It shows that whatever the error of the profiler is, the learning time remains independent of it and happens to be 3 days for all of them. On the other hand, the profiling error has a strong impact on the estimation error of the network. It is clear from Fig. 11b, which depicts the overall RMSE of the network with each considered profiler over all operational days, excluding the first three training days. Comparing these results with the individual trace based learning in Fig. 6, indicate that the profiler accuracy under 10% error will be acceptable in this approach.

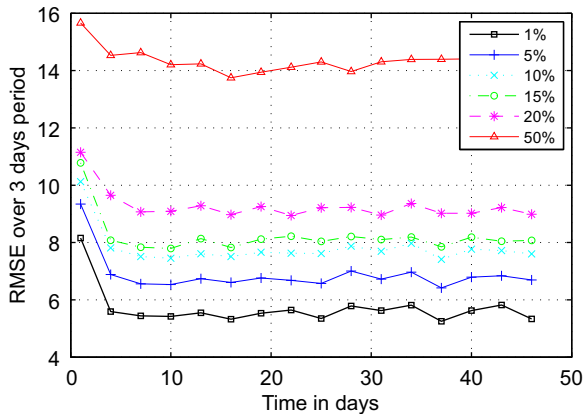


Fig. 10. Evolution of error.

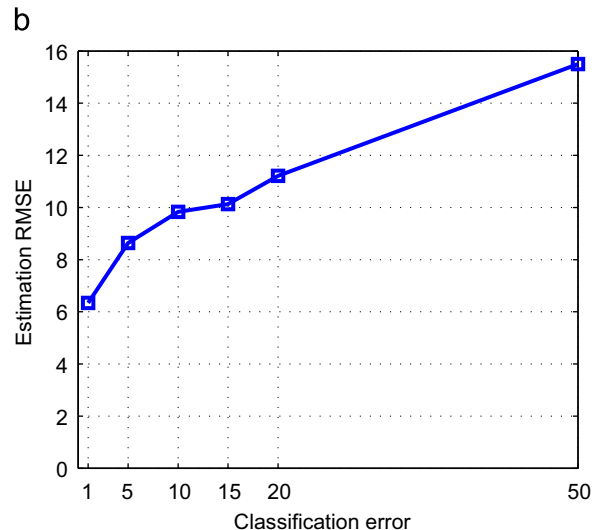
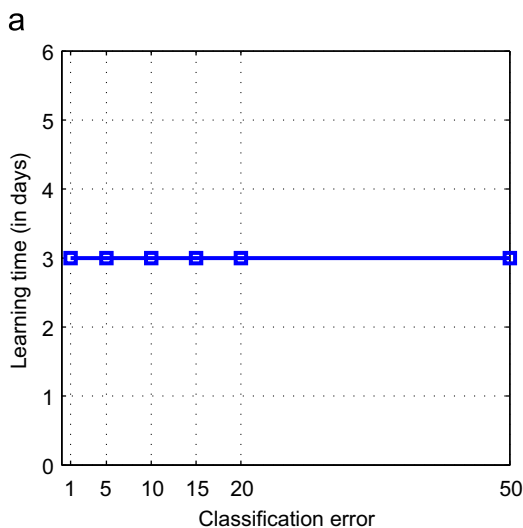


Fig. 11. Impact of classification error.

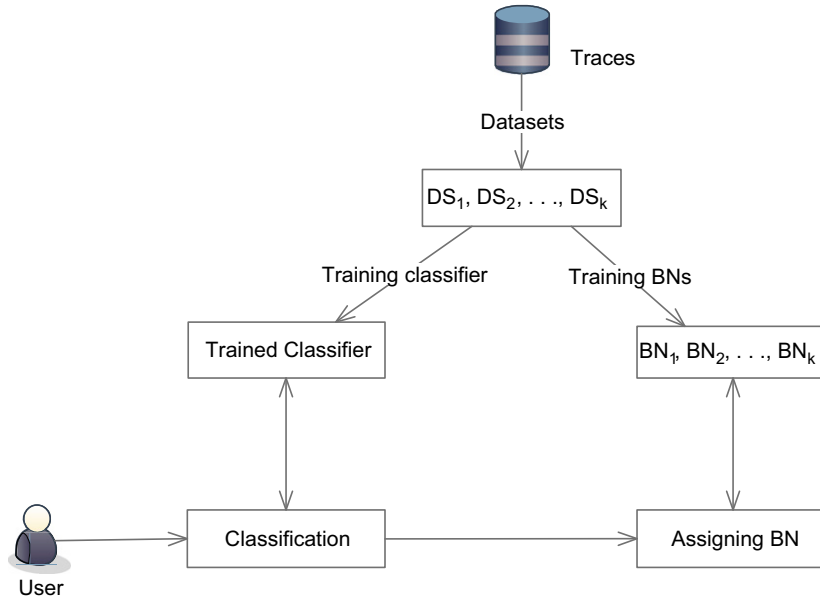


Fig. 12. Coupling a classifier with trained BNs.

This approach shows a significant improvement over the previous one, where initially a 40 days of traces were required on average for learning the behavior while now it has been reduced to 3 days. Collecting individual traces incurs an overhead, however, it can be minimized by requiring a peer to collect the traces once after joining. Moreover, a limit on the collected amount of traces can be put on each peer, since once fully trained, no extra traces are required to accelerate the learning process. After this initial duration of a new comer, only the local traces are sufficient to update the knowledge base of the network.

This learning period can be further reduced and the requirement of collecting traces of the similar behaviors can be eliminated through the combination of pre-trained Bayesian networks and a classifier. We depict the general idea in Fig. 12.

In this approach, first user classes (personas) must be identified. It might be done through analyzing existing traces of users collected from a P2P live video streaming system in function. These traces are divided into datasets, each of them corresponding to a user class. A dedicated BN is trained from dataset of each group and a classifier is learned from all datasets of all groups. As a user arrives, the classifier classifies him into a group and the appropriate BN is assigned to that user. All the required estimations are then carried out by the assigned BN throughout the online period of the user.

We do not consider dynamic behaviors in which the current behavior of a user may be different from the past one. In that case relevance feedback mechanisms [43] can be used.

5. Applications

Our Bayesian network can perform a multiway inference that makes possible the estimation of all modeled

variables. These estimations have several applications such as anticipation of churn, users' arrival/departure rates, stability and popularity dynamics. Here we evaluate our Bayesian network over the anticipation of users' departures, popularity dynamics, and arrival/departure rates. In case of departures anticipation, we also compare the results of our model with those of two other estimators.

5.1. Departure anticipation

The current session duration of a user can be predicted through our network, which determines the departure time and stability of a user. This estimation can help to optimize a P2P live streaming system for improved performance. In [14], we proposed a non-contextual approach to anticipate a peer's departure. This approach does not take into account any contextual information. Here, we compare the accuracy of our Bayesian network to those non-contextual estimators over departure anticipation. The non-contextual approach provides different estimators based on EMA (Exponential Moving Average) and Bayes' rule (BR). We briefly discuss these estimators.

5.1.1. EMA and BR

EMA is a statistical technique that estimates an average from a set of values by giving exponentially decreasing weights to older values. As given in (2), ES_t is the current session duration, S_{t-1} is the actual duration of the previous session, ES_{t-1} is the length of the last estimated session and $\alpha \in [0, 1]$ is a weighting factor:

$$ES_t = \alpha \times S_{t-1} + (1 - \alpha) \times ES_{t-1} \quad (2)$$

This estimator was further extended through reducing the estimated session by 20% for increasing the number of

successful anticipations. This estimator produced better results than the normal one, therefore here we choose the same for comparison. In the sequel, we will refer to it by EMA.

The second non-contextual estimator is based on the Bayes' Rule (BR). It is a probabilistic approach that enables the estimation of a peer's current session duration from observations of past sessions. Let $T \in \mathbb{R}$ be the maximum possible session duration of a node A . We discretize T in k equal time interval $\{t_1, t_2, \dots, t_k\}$ such that $t_1 < t_2 < \dots < t_{k-1} < t_k$. For each time interval $[t_i, t_{i+1}]$ we define a binomial variable α_i . The prior probability of the event ϕ_j , meaning that the next session duration S_i will be at least equal to t_j , can be modeled by the Dirichlet density function as given in (3a) where α_j is the number of observations when $S_i \geq t_j$. The estimated prior probability of $(S_i \geq t_j)$ is given by (3b) where $|O|$ is the set of already observed session durations. Let us notice that the variables α are not independent. So we define a mechanism to update the $\langle \alpha \rangle$ vector. After observing a set O' of some new session durations, the posterior probability of a session duration S_i to be at least equal to t_j is computed through (3c) where $o_j \in O$ and $|o_j|$ is the number of observations where $S_i \geq t_j$.

$$f(\phi_1 \dots \phi_{k-1} | \alpha_1 \dots \alpha_k) = \frac{\Gamma(\alpha_1 + \alpha_2 + \dots + \alpha_k)}{\Gamma(\alpha_1)\Gamma(\alpha_2) \dots \Gamma(\alpha_k)} (\phi_1^{\alpha_1-1} \phi_2^{\alpha_2-1} \dots \phi_k^{\alpha_k-1}) \quad (3a)$$

$$f(\phi_j) = \frac{\alpha_j + 1}{|O| + 2} \quad (3b)$$

$$f(\phi_j | O, O') = \frac{\alpha_j + |o_j| + 1}{|O| + |O'| + 2} \quad (3c)$$

This estimator updates the list of posterior probabilities corresponding to each t_j at the completion of each session. The estimation of a current session duration t_i is performed with Eq. (4) in which the probability threshold

P_{th} should be fixed:

$$t^* = \max_{t_j} (f(\phi_j) \geq P_{th}) \quad (4)$$

5.1.2. Comparison metrics

We compare EMA and BR with our Bayesian network (BN) according to the following metrics. First, we measure *success* which stands for the number of times our estimated session duration is less than or equal to the actual session duration. In this case a client peer is able to successfully anticipate the departure of its content provider peer and react proactively. Second, we measure the *early reaction time* (ERT) for all successful estimations. This measurement shows how optimal the anticipation is. Third, we measure the error in terms of RMSE in all cases, which reflects the overall accuracy of estimators. As previously, we discuss detailed results of all these estimators in the two deployment scenarios.

5.1.3. Global case

Since the accuracy of EMA and BR changes with variation in the values of their parameters (α and P_{th}), first we evaluate these two estimators through varying these two parameters and analyzing their success and RMSE.

We plot the α versus success for EMA and P_{th} versus success for BR in Fig. 13. The success of both EMA and BR increases with an increase in the value of α and P_{th} respectively, varying smoothly from 50% to 64% for EMA, while for BR it reaches from about 11% to 95%.

On the other hand, the RMSE of EMA slightly increases with an increase in the value of α as shown in Fig. 14. Here, the error variation of BR is quite interesting. It starts at the highest for the lowest used threshold, decreases with an increase in the threshold, reaching to minimum at 0.5 and then starts climbing again. It means that BR over- and under-estimates sessions by a large margin, resulting in inconsistent behavior over different probability thresholds. Concerning BN, it produces a 46.17% success with an RMSE of 19.14 min in this scenario.

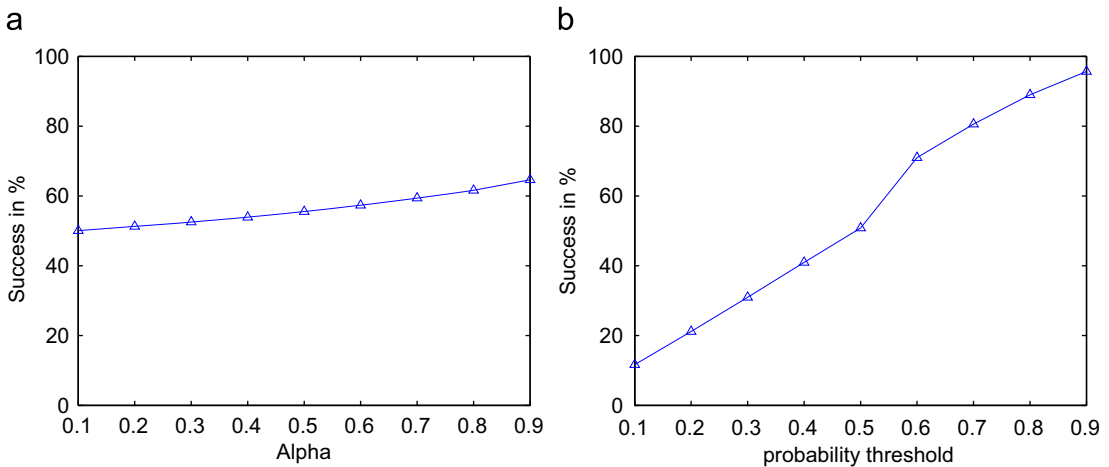


Fig. 13. Evolution of success in global case: (a) EMA; (b) BR.

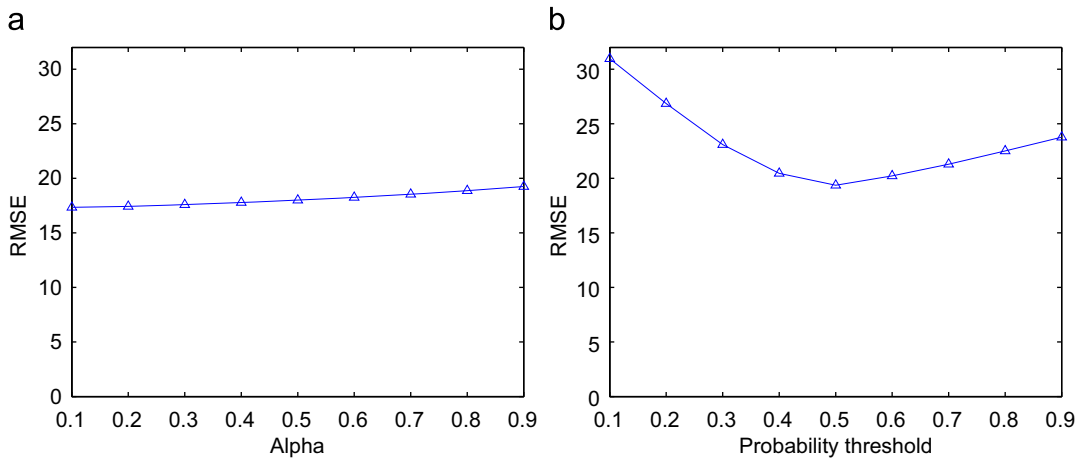


Fig. 14. Evolution of error in global case: (a) EMA; (b) BR.

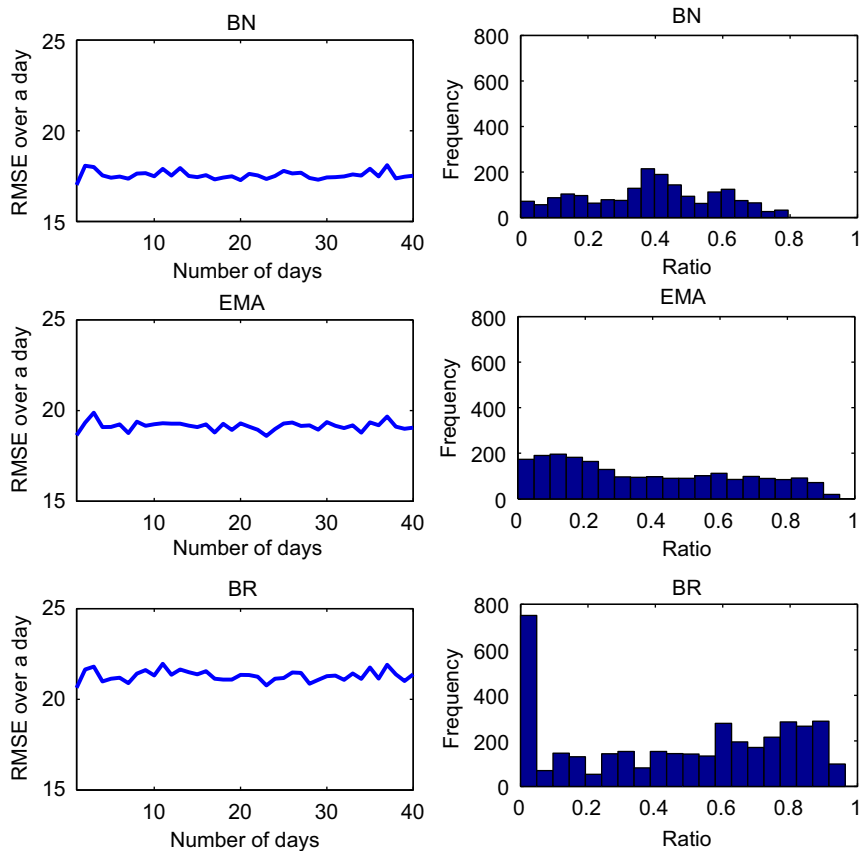


Fig. 15. Comparison: (a) RMSE; (b) ERT ratios.

To choose the best parameters of these estimators for further detailed results and a comparison with the BN, we set P_{th} to 0.5 in case of BR, with which it produces the minimum error while the success varies too much. Similarly, in case of EMA, we set α to 0.9, with which it achieves the highest success with a slight increase in the error.

We depict RMSE of all the three estimators in Fig. 15a. Here, BN generates a slightly smaller error than other estimators. However, the difference in error of all estimators is quite low and the magnitude of error is high for all. As shown in Fig. 15b, BR produces slightly better results in terms of ERT ratios to actual session durations, where a significant portion of ratios lies near to zero. However,

ratios are spread over the entire range, which shows an inconsistency.

Overall, EMA and BR show better results in this scenario, than BN. Between these two, EMA produces higher success than BR over the considered values of α and P_{th} . However, BR produces better ERT ratios. Hence, none of these estimators emerge as a clear winner in this scenario.

5.1.4. Local case

As we did in the global case, in the local case we evaluate and compare the accuracy of the three estimators over the anticipation of peers' departures. Before discussing results of BN, we plot the success of EMA and BR over different values of α and P_{th} in Fig. 16. Same as in the global case, here too, EMA shows a slight increase in success with an increase in the value of α , while BR shows a larger increase in success with an increase in the value of P_{th} .

Similarly, the error shown in Fig. 17, evolves in the same way as in the global case. Concerning BN, it successfully

anticipates 50% of departures, with an RMSE of 6.5 min. These numbers show a good accuracy, however, the lower success arises from the fact that even a slightly over-estimated session duration results into an unsuccessful anticipation. The RMSE of BN being quite lower than the other estimators, we reduce its estimation by 20%, as we did for EMA, to lower these slightly over-estimated values. As a result, the success reaches 84.4%. On the other hand, the RMSE also increases reaching up to 8.2 min. However, it is still far smaller than those generated by EMA and BR.

To further compare EMA and BR with BN, we choose their parameters with the best results. Here both α and P_{th} are set to 0.9. The average RMSE over 10 days period for each persona is shown in Fig. 18. Here, one can notice that BN consistently produces the least error among all estimators, while EMA and BR not only produce larger errors but they are also not consistent over different personas. For instance, BR produces an RMSE of just under 40 for persona 'J', while under 20 for 'S'. Similarly, in the case of

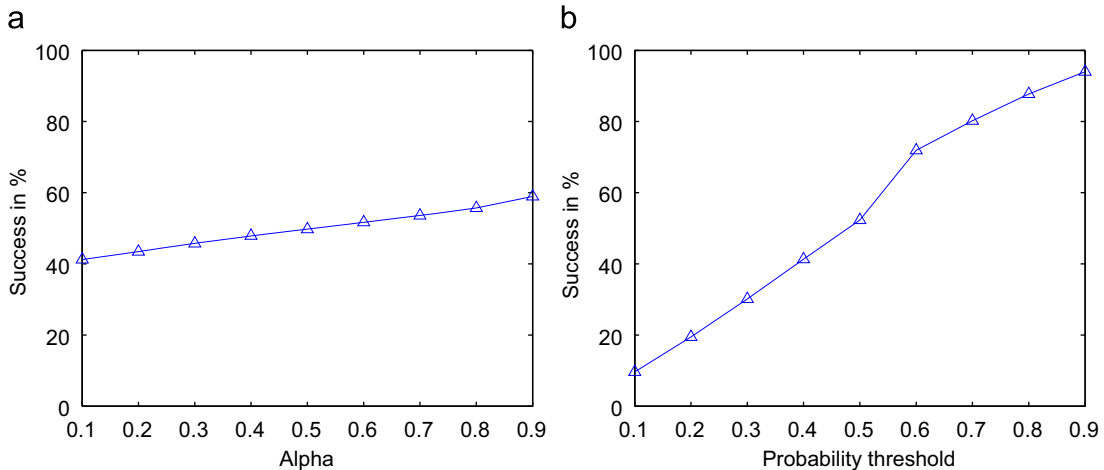


Fig. 16. Evolution of success in local case: (a) EMA; (b) BR.

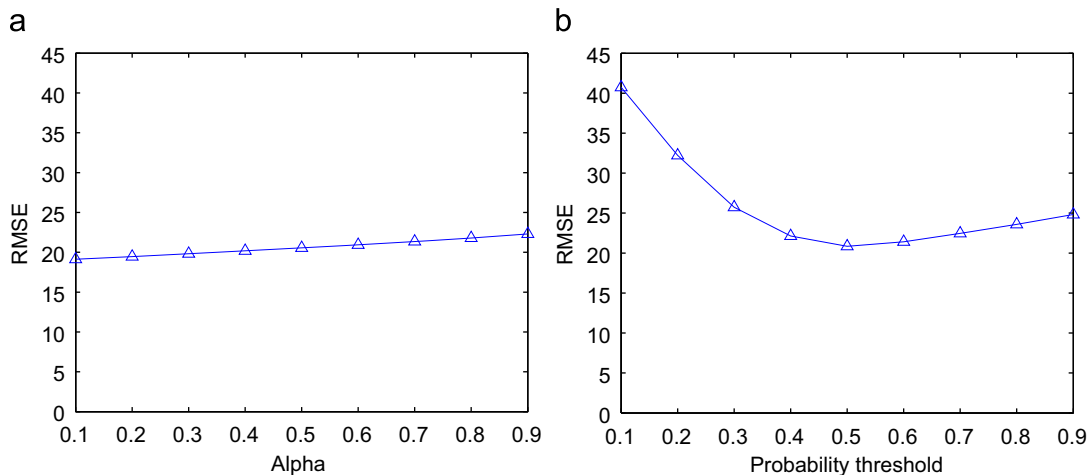


Fig. 17. Evolution of error in local case: (a) EMA; (b) BR.

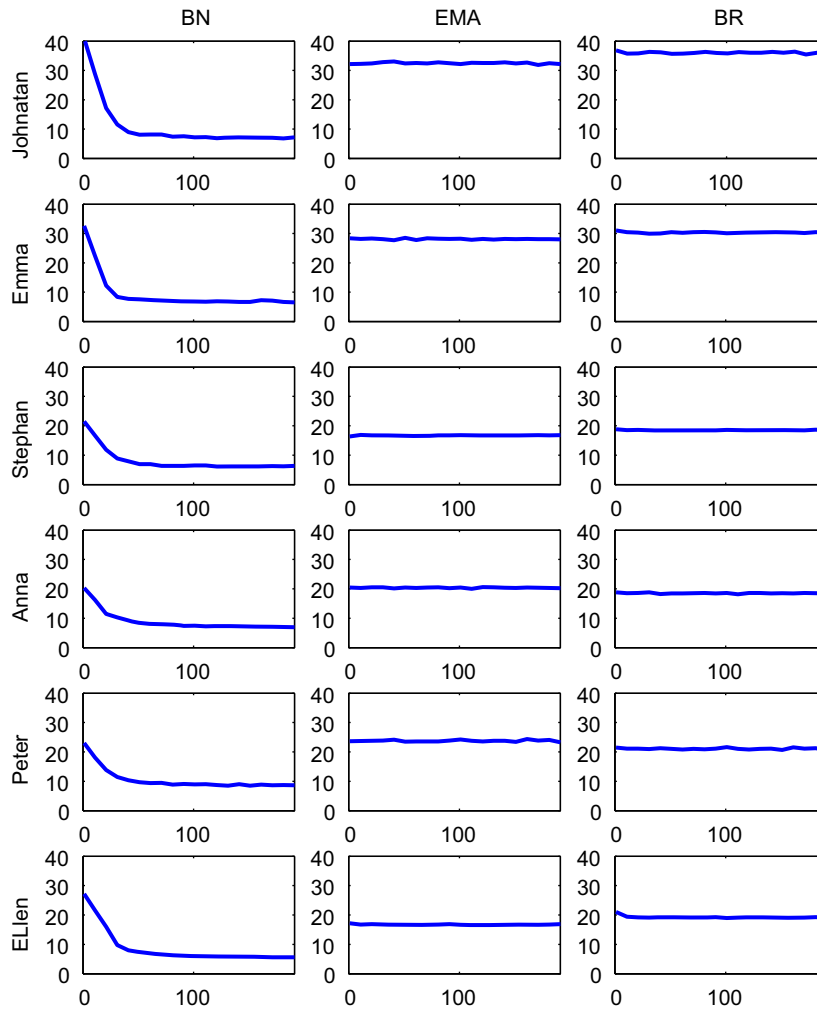


Fig. 18. RMSE averaged over a 10 days period (number of days for each persona is given on the x-axis and averaged RMSE is given on the y-axis).

EMA, it remains above 30 and around 15 for the same personas respectively. By contrast, BN consistently produces an RMSE of less than 10 after training.

We also depict the ERT ratios of all the three estimators for all personas in Fig. 19. Here, again BN produces the lowest ratios consistently showing a good accuracy, while BR and EMA produce larger ratios spread over all the range. These results indicate the inefficiency of non-contextual models since they struggle to accurately estimate sessions of the same user with consistency. For instance, ratios of EMA show two groups, one around 0.2 and another around 0.8. Similarly, BR has a group of small ratios, then another one in which most of the ratios lie in the range of 0.6–0.9. By contrast, BN consistently produces smaller ratios indicating a good accuracy over all personas.

Looking at the consistency over different personas, again BN shows quite similar results for all of them. On the other hand, BR is highly inconsistent while EMA shows a better similarity but its success is far lower than BN and its errors are higher. Clearly, all these results show that BN outperforms EMA and BR in the local case.

5.2. Estimation of system-level variables

As a second application of our Bayesian network, we show how it can be used as an indirect measurement of the network size and dynamics. Being able to anticipate the network state could help in system dimensioning and especially flash crowds management. This section only gives a proof of feasibility of this kind of estimation and obviously further investigations would be required to fully validate it. Especially a direct comparison with protocols dedicated with such an objective would be necessary to estimate the relative performance of our indirect measurement but such a work is out of the scope of this paper.

Churn [44] in P2P networks is the independent arrival and departure of users. It has a strong impact on the performance of P2P live video streaming systems. A high arrival rate leads to failures and early departures from users. It is due to the random joining mechanism where newly arrived peers rely on other peers who have also recently joined the system and have not received sufficient amount of stream to transmit. Consequently startup

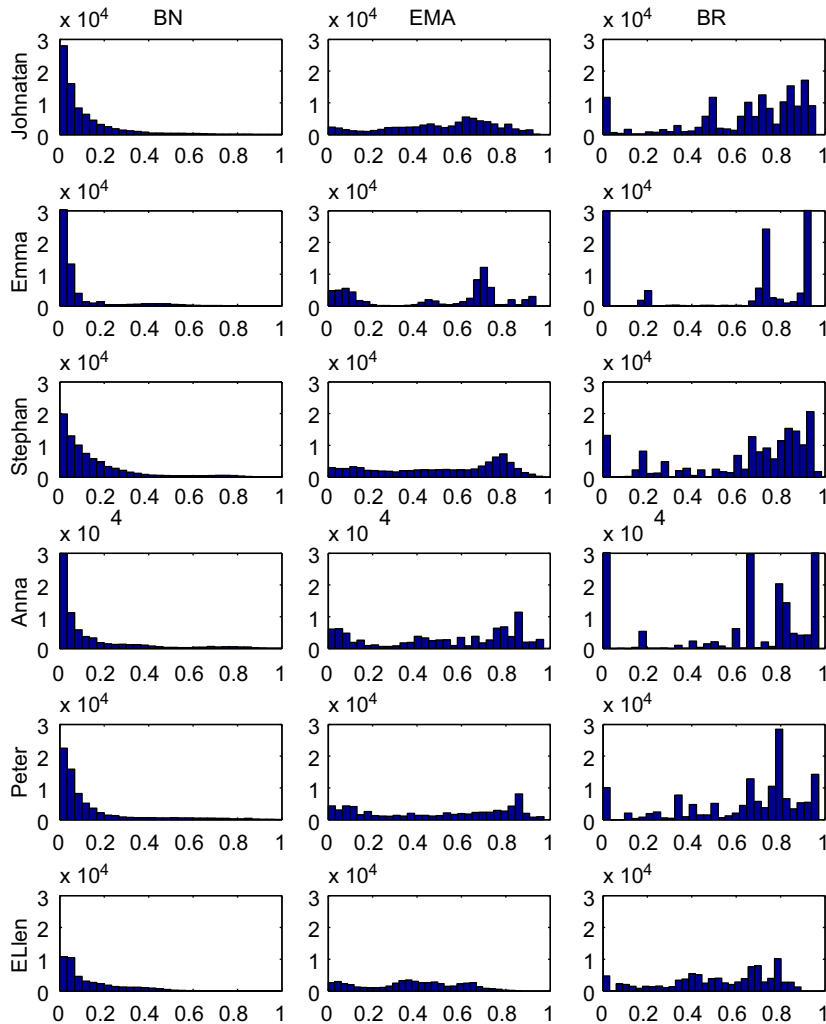


Fig. 19. Histograms of ERT ratios (ratios are given on the x-axis and their frequencies are given on the y-axis).

delays are increased. High departure rates also disrupt the stream to downstream peers. Therefore, the collective behavior of user arrivals and departures called churn is important towards the optimization of P2P streaming system.

The number of arrivals and departures alone cannot accurately determine the impact of churn on the system. The size of the system or population should also be considered. Our Bayesian network is able to predict these three variables from which a churn level can be deduced. Here we only present the estimation of these three variables. As mentioned earlier, global scenario can be useful for the estimation of system-level variables, thus we utilize the same scenario here. To estimate arrival rate, departure rate and popularity in this scenario, we hide these three variables and the session duration which is unknown for each user while he is online. We let our network estimate them on the arrival of each user and compare the actual values with the estimated ones.

We depict the RMSE of estimations taken over each day in Fig. 20a. One can notice that errors for arrival and

departure rates are less than those for popularity. The reason is that popularity depends upon the arrival and departure rates and hence an error in these values results in an increase in the error of popularity estimation. To further evaluate the accuracy, we depict histograms of absolute errors for all estimations in Fig. 20b. Here, again a large portion of estimations lie around zero which is a good accuracy keeping in mind the four variables in the network were unknown. To further elaborate these estimations, we also depict a scatter plot of average versus estimated values for the first 5 days in Fig. 20c. It shows a good accuracy in the given context.

All these results show that this network can be used for making decisions at the provider side to allocate resources dynamically. This will enable a better utilization of resources.

6. Conclusion and future perspectives

Due to their architectural models in which the content is relayed from user to user, P2P video streaming systems

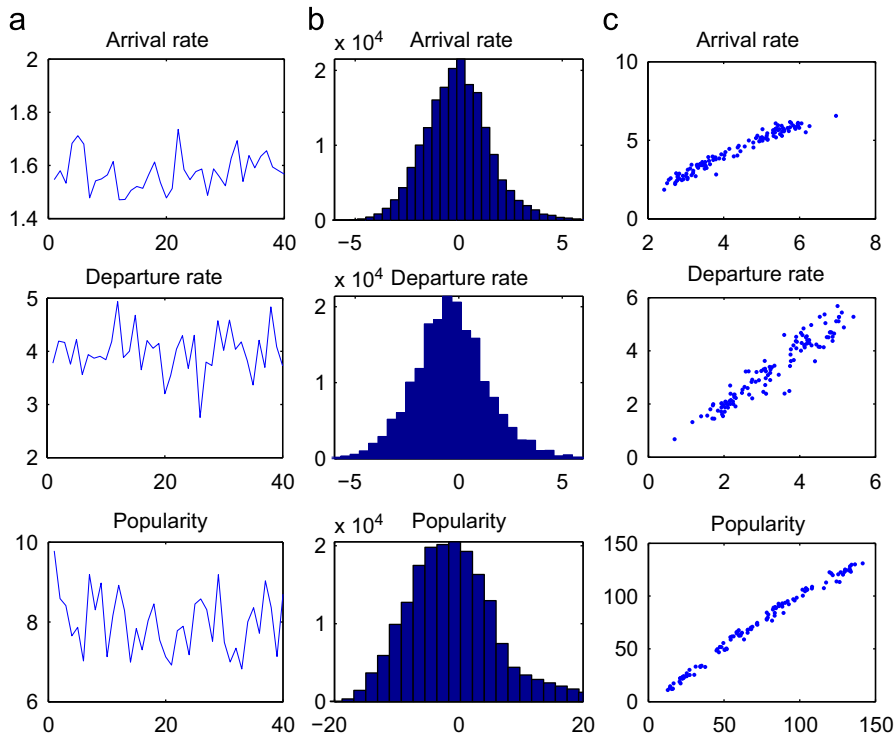


Fig. 20. Estimation of system-level variables: (a) Days on the x-axis, RMSE on the y-axis. (b) Absolute error on the x-axis, frequencies on the y-axis. (c) Actual on the x-axis, estimated on the y-axis.

can largely benefit from the integration of user behavior in their operational mechanisms. In this paper, we presented a Bayesian network model, its evaluation and improvement in its learning process. We also compared the accuracy of this model with that of non-contextual approaches.

The Bayesian network we proposed is a contextual model that includes all the involved variables of user behavior found in large-scale measurement studies over live video streaming systems. It is able to estimate any of these variables which enables to make multiple decisions towards improving the streaming quality. These variables are of two types, peer-level which can get different values for each user and system-level which are the same for the whole community.

We simulated two deployment scenarios of the Bayesian network namely global and local. The former requires only one network for all users while the latter requires a dedicated network for each user. We evaluated the performance of the network in both of these scenarios. While estimating the session duration of individual users which is a peer-level variable, the globally deployed network does not produce accurate results. However, this network can be useful for the estimation of global parameters such as the population size. By contrast, using a dedicated Bayesian network for each individual user produces very accurate results for the estimation of peer-level variables. Concerning the performance of our network, we proposed an aggregation-based mechanism that accelerates its learning rate. It showed a significant improvement in learning period which is reduced from

about 40 to 3 days. We also proposed a blueprint of a classifier-based approach that can be coupled with trained Bayesian network to eliminate the need of traces collection and individual learning.

In order to understand how our Bayesian network would operate in concrete applications, we used it to infer an estimation of session duration, population and arrival/departure rates. The estimation of session duration aims at anticipating the departure of peers and determines their stability. It is a crucial factor in P2P live streaming systems and that is why in this case, we also compared the results of our network with those of two other estimators. Results indicate that our Bayesian network outperforms the non-contextual estimators in estimation of the session duration. It shows the suitability of contextual approach to this kind of problems. Moreover, results concerning the estimation of system-level metrics show a good accuracy, which increases the range of its applications.

This work opens numerous perspectives. Currently, we are working on the model of a classifier for users' profiling, which will be first used for enhancing the learning rate but which also covers more extended applications such as the building of user-profiled P2P topologies. Such an application could enable the placement of peers in the topology according to their profile enabling thus more efficient topologies. Furthermore, we are working on the design and implementation of a protocol that based on the estimations of the Bayesian network, will optimize the P2P live streaming system for improved users' Quality-of-Experience (QoE).

References

- [1] I. Ullah, G. Doyen, G. Bonnet, D. Gaïti, User Behavior Anticipation in P2P Live Video Streaming Systems through a Bayesian Network, in: Twelfth IFIP/IEEE International Symposium on Integrated Network Management, IEEE, 2011, pp. 337–344.
- [2] K. Xu, M. Zhang, M. Ye, D.M. Chiu, J. Wu, Identify P2P traffic by inspecting data transfer behavior, *Computer Communications* 33 (2010) 1141–1150. ISSN 0140-3664.
- [3] N.D. Doulamis, P.N. Karamolegkos, A. Doulamis, I. Nikolakopoulos, Exploiting semantic proximities for content search over p2p networks, *Computer Communications* 32 (5) (2009) 814–827. ISSN 0140-3664.
- [4] D. Leonard, Z. Yao, X. Wang, D. Loguinov, On static and dynamic partitioning behavior of large-scale P2P networks, *IEEE/ACM Transactions on Networking* 16 (6) (2008) 1475–1488. ISSN 1063-6692.
- [5] Q. Feng, Y. Wu, Y. Sun, J. Jiang, Y. Dai, User behavior modeling in peer-to-peer file sharing networks: Dissecting download and removal actions, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), ISSN 1520-6149, 2009, pp. 3477–3480.
- [6] X. Hei, C. Liang, J. Liang, Y. Liu, K.W. Ross, A measurement study of a large-scale P2P IPTV system, *IEEE Transactions on Multimedia* 9 (8) (2007) 1672–1687.
- [7] S. Xie, G.Y. Keung, B. Li, A Measurement of a large-scale peer-to-peer live video streaming system, in: International Conference on Parallel Processing Workshops, 2007, pp. 57–62.
- [8] J. Jia, C. Li, C. Chen, Characterizing PPStream across Internet, in: IFIP International Conference on Network and Parallel Computing Workshops, 2007, pp. 413–418.
- [9] C. Li, C. Chen, Measurement-based study on the relation between users' watching behavior and network sharing in P2P VoD systems, *Computer Networks* 54 (2010) 13–27. ISSN 1389-1286.
- [10] Z. Liu, C. Wu, B. Li, S. Zhao, Distilling superior peers in large-scale P2P streaming systems, in: IEEE INFOCOM, 2009, pp. 82–90.
- [11] Y. Tang, L. Sun, J.-G. Luo, Y. Zhong, Characterizing user behavior to improve quality of streaming service over P2P networks, in: Advances in Multimedia Information Processing, 2006, pp. 175–184.
- [12] F. Wang, J. Liu, Y. Xiong, Stable peers: existence, importance, and application in peer-to-peer live video streaming, in: IEEE INFOCOM, 2008, pp. 1364–1372.
- [13] S. Horovitz, D. Dolev, Collabrium: active traffic pattern prediction for boosting P2P collaboration, in: IEEE International Workshops on Enabling Technologies, 2009, pp. 116–121.
- [14] I. Ullah, G. Bonnet, G. Doyen, D. Gaïti, Improving Performance of ALM Systems with Bayesian Estimation of Peers Dynamics, in: Twelfth IFIP/IEEE International Conference on Management of Multimedia and Mobile Networks and Services, MMNS 2009, Springer, 2009, pp. 157–169.
- [15] S. Spoto, R. Gaeta, M. Grangetto, M. Sereno, Analysis of PPLive through active and passive measurements, in: IEEE International Symposium on Parallel & Distributed Processing, ISBN 978-1-4244-3751-1, 2009, pp. 1–7.
- [16] L. Vu, I. Gupta, J. Liang, K. Nahrstedt, Measurement and modeling of a large-scale overlay for multimedia streaming, in: International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, 2007, pp. 1–7.
- [17] T. Silverston, O. Fourmaux, A. Botta, A. Dainotti, A. Pescapè, G. Ventre, K. Salamati, Traffic analysis of peer-to-peer IPTV communities, *Computer Networks* 53 (4) (2009) 470–484.
- [18] K. Shami, D. Magoni, H. Chang, W. Wang, S. Jamin, Impacts of peer characteristics on P2PTV networks scalability, in: IEEE INFOCOM, 2009, pp. 2736–2740.
- [19] X. Zhang, J. Liu, B. Li, On large scale peer-to-peer live video distribution: coolstreaming and its preliminary experimental results, in: IEEE International Workshop on Multimedia Signal Processing, 2005.
- [20] B. Li, S. Xie, G.Y. Keung, J. Liu, I. Stoica, H. Zhang, X. Zhang, An empirical study of the coolstreaming + system, *IEEE Journal on Selected Areas in Communications* 25 (9) (2007) 1627–1639.
- [21] S. Agarwal, A case study of large scale P2P video multicast, in: International Conference on IP Multimedia Subsystem Architecture and Applications, 2007, pp. 1–5.
- [22] Y. Tang, L. Sun, K. Zhang, S.-Q. Yang, Y. Zhong, Longer, better: on extending user online duration to improve quality of streaming service in P2P networks, in: IEEE International Conference on Multimedia and Expo, 2007, pp. 2158–2161.
- [23] K. Sripanidkulchai, B. Maggs, H. Zhang, An analysis of live streaming workloads on the Internet, in: IMC' 04, ISBN 1-58113-821-0, 2004, pp. 41–54.
- [24] E. Veloso, V. Almeida, W.J. Meira, A. Bestavros, S. Jin, A hierarchical characterization of a live streaming media workload, *IEEE/ACM Transactions on Networking* 14 (1) (2006) 133–146.
- [25] T. Qiu, Z. Ge, S. Lee, J. Wang, J. Xu, Q. Zhao, Modeling user activities in a large IPTV system, in: 9th ACM SIGCOMM Conference on Internet Measurement, 2009, pp. 430–441.
- [26] T. Qiu, Z. Ge, S. Lee, J. Wang, Q. Zhao, J. Xu, Modeling channel popularity dynamics in a large IPTV system, in: Eleventh International Joint Conference on Measurement and Modeling of Computer Systems, 2009, pp. 275–286.
- [27] M. Cha, P. Rodriguez, J. Crowcroft, S. Moon, X. Amatriain, Watching television over an IP network, in: Eighth ACM SIGCOMM Conference on Internet measurement, ISBN 978-1-60558-334-1, 2008, pp. 71–84.
- [28] Z. Liu, C. Wu, B. Li, S. Zhao, Why are peers less stable in unpopular P2P streaming channels?, in: Networking, 2009, pp. 274–286.
- [29] B. Li, S. Xie, Y. Qu, G.Y. Keung, C. Lin, J. Liu, X. Zhang, Inside the new coolstreaming: principles, measurements and performance implications, in: IEEE INFOCOM, 2008, pp. 1031–1039.
- [30] J. Pearl, S. Russell, Bayesian Networks, Technical Report, University of California, Los Angeles, CA 90095, 2000.
- [31] G. Bonnet, I. Ullah, G. Doyen, L. Fillatre, D. Gaïti, I. Nikiforov, A semi-Markovian individual model of users for P2P video streaming applications, in: Forth IFIP International Conference on New Technologies, Mobility and Security (NTMS), IEEE, ISSN 2157-4952, 2011, pp. 1–5.
- [32] A. Rudstrom, M. Sjolinder, Capturing TV User Behaviour in Fictional Character Descriptions, Technical Report, Swedish Institute of Computer Science, 2008.
- [33] P. Branch, G. Egan, B. Tonkin, Modeling interactive behaviour of a video based multimedia system, in: Proceedings of the IEEE International Conference on Communications, 1999, pp. 978–982.
- [34] M. Vilas, X.G. Pañeda, R. García, D. Melendi, V.G. García, User behaviour analysis of a video-on-demand service with a wide variety of subjects and lengths, in: Thirty-First EUROMICRO Conference on Software Engineering and Advanced Applications, 2005, pp. 330–337.
- [35] H. Yu, D. Zheng, B.Y. Zhao, W. Zheng, Understanding user behavior in large-scale video-on-demand systems, *ACM SIGOPS Operating Systems Review* 40 (4) (2006) 333–344.
- [36] A. Brampton, A. MacQuire, I. Rai, N.J.P. Race, L. Mathy, M. Fry, Characterising user interactivity for sports video-on-demand, in: International Workshop on Network and Operating Systems Support for Digital Audio & Video, 2007.
- [37] S. Acharya, B. Smith, Characterizing user access to videos on the world wide web, in: Multimedia Computing and Networking, 2000, pp. 130–141.
- [38] B. Chang, L. Dai, Y. Cui, Y. Xue, On feasibility of P2P on-demand streaming via empirical VoD user behavior analysis, in: Twenty-Eighth International Conference on Distributed Computing Systems Workshops, 2008, pp. 7–11.
- [39] E. Veloso, V. Almeida, W. Meira, A. Bestavros, S. Jin, A hierarchical characterization of a live streaming media workload, in: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement, 2002, pp. 117–130.
- [40] L. Massoulié, E.L. Merrer, A.-M. Kermarrec, A. Ganesh, Peer counting and sampling in overlay networks: random walk methods, in: Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing, PODC '06, ACM, 2006, pp. 123–132.
- [41] A. Montresor, A. Ghodsi, Towards robust peer counting, in: Ninth International Conference on Peer-to-Peer Computing, IEEE, 2009, pp. 143–146.
- [42] A. Binzenhöfer, K. Leibnitz, Estimating churn in structured P2P networks, in: Twentieth International Teletraffic Conference on Managing Traffic Performance in Converged Networks, ITC20'07, Springer-Verlag, 2007, pp. 630–641.
- [43] N. Doulamis, A. Doulamis, Evaluation of relevance feedback schemes in content-based in retrieval systems, *Signal Processing: Image Communication* 21 (4) (2006) 334–357.
- [44] D. Stutzbach, R. Rejaie, Understanding churn in peer-to-peer networks, in: Sixth ACM SIGCOMM Conference on Internet Measurement (IMC 2006), 2006, pp. 189–202.