

Towards User-aware Peer-to-Peer Live Video Streaming Systems

Ihsan Ullah and Guillaume Doyen and Dominique Gaiti
ERA / Institut Charles Delaunay – UMR 6279
Université de Technologie de Troyes
12 rue Marie Curie – 10010 TROYES – France
Email: ihsan.ullah.2012,doyen,gaiti@utt.fr

Abstract—The involvement of end hosts to relay content to each other makes Peer-to-Peer (P2P) video streaming systems potentially scalable and easy to deploy with low cost. Since end hosts are controlled by users, their intermittent presence directly impacts the performance of these systems. Current approaches do not consider and integrate user behavior in their design, hence suffering from Quality-of-Service (QoS) issues. In this thesis, we focus on the user behavior from its understanding to modeling and integrating into P2P systems. We propose an autonomous management framework that (1) enables peers to learn the behavior of users and organize themselves in such a way to minimize its impact and improve the streaming quality, and (2) controls the topology of push-based systems through a stabilization process that continuously pushes unstable peers towards the leaves of the tree. We evaluate and compare these models through both simulations and experimentations over PlanetLab which show that the integration of user behavior into P2P systems considerably improves their performance.

I. INTRODUCTION

Video streaming is becoming increasingly popular among the Internet users. Enabling such a service is challenging due to large bandwidth requirements to deliver content to a large number of users. IP multicast is the natural way to enable such a service but it could not be deployed at the Internet scale. At the other hand, Content Delivery Networks (CDNs) and centralized Client/Server (C/S) architecture are expensive solutions. Peer-to-Peer (P2P) approach has emerged as an excellent alternative due to its low cost and ease of deployment.

P2P systems form an overlay through making direct links among end-hosts. End-hosts in these systems are termed as peers or nodes. A peer in such a network receives contents and transmits it to other peers. To deliver stream to all peers in the network, currently proposed systems adopt different strategies. They can be broadly classified into three major types namely, push-based, pull-based and hybrid push/pull approach. In push-based strategy [1], end-hosts are organized into well-defined parent children relationships where a parent peer always relays the content to its child peers as it receives. These systems are also called tree-based systems. By contrast, pull-based systems [2] do not form a fixed delivery structure. A peer in such a system establishes links with several other peers to form partnerships. A peer informs its partners about the content availability through advertising its buffer maps. Other peers pull the content from partner peers through explicit requests. Hybrid systems [3] operate both in push and pull mode at the same time. Peers form a tree structure to receive the content

through push operations while to fetch missing content through pull operation, they form links with other peers.

Nevertheless, P2P systems for streaming applications face challenges to provide a comparable quality to traditional television. These systems are highly dependent on end-hosts which are controlled by users. These latter join and leave the system independently, a phenomenon called churn, which turns the P2P network highly dynamic. Especially, push-based systems are highly churn-sensitive. A failure or abrupt departure of a peer from the system badly impacts the stream continuity to its downstream peers. Current approaches deal with the system dynamics indirectly through making the network topology dynamics-resilient and by using large buffers. Although these strategies attempt to ensure streaming continuity, they induce Quality-of-Service (QoS) issues impacting the Quality-of-Experience (QoE) perceived by users. Such problems include longer delays (startup/channel switching, playback) and playback lags among the peers. For instance, users wait tens of seconds for the video playback to start after they join a channel or switch from another one. Moreover, they watch channels with minutes of playback lags among peers [4]. All these problems result in users' dissatisfaction since they are habitual with traditional television and expect a similar service on the Internet. Therefore, the behavior of P2P streaming users becomes crucial to provide a comparable service to traditional television.

In this thesis, we focus on the user behavior of P2P live video streaming systems. Especially, we propose to integrate it into P2P systems to make them user-aware. To reach that aim, we propose an autonomous management framework that enables peers to anticipate the behavior of users and self-organize the network for improved streaming quality. We get insight into user behavior through a complete synthesis of measurement studies collected from the literature. Based on this synthesis, we propose a few models that predict user activities and especially the estimation of a peer departure time. This estimation is then used to enable peers in arranging themselves in such a way to minimize the effect of a peer departure, thus reducing the global churn impact. We perform several simulations and experiments on PlanetLab to validate our approach.

The rest of this document is structured as follows. Section II presents related work. Section III demonstrates our autonomous topology management framework that integrates user behavior into P2P streaming systems for performance improvement. Section IV describes validation of our approaches

through simulations and experiments. Finally, section V concludes and gives future directions.

II. RELATED WORK

Numerous works on user behavior in video streaming applications are based on measurement studies [5], [6], [7], [8]. These measurements provide useful information towards understanding user behavior, but no single study gathers and analyzes all the concerned elements. Moreover, measurement results depend upon the system and experimental conditions which does not permit to use a single study as a basis for behavior modeling. Apart from measurements, some works also propose user behavior models. We briefly mention the main ones here.

Tang *et al.* [9] observe through measurements that users stay longer watching a channel if they have been holding that channel since long. Based on this observation, they propose a peer joining mechanism which prefers to choose a peer with longer elapsed time as an upstream peer. Wang *et al.* [10], [3] adopt an identical approach to extract stable peers and place them in the backbone of the network. Liu *et al.* [11] determine peers' resilience through their age to reduce service disruption.

Zheng *et al.* [12] observe that service quality is degraded with an increase in node churn. They present a strategy for peer selection that considers the availability of content to balance connection efficiency and stability requirements. This work establishes a correlation between node churn and service quality, however, it does not propose an accurate estimation method of the former.

Liu *et al.* [13] find through a measurement study that peers' stability and their bandwidth contribution are impacted by several factors such as a user watching a popular channel who received a good initial streaming quality tends to stay longer. They also propose statistical models for estimating the stability and bandwidth contribution of peers.

However, the above-mentioned models suffer from the following limits. Firstly, they are incomplete, as they do not consider all the involved variables of user behavior. Their type of models address one issue while ignoring others. Secondly, the models are global, thus assuming that all users behave in a similar way. This is not always the case since individuals have different habits and preferences. Finally, they do not consider all the dependency relationships between user behavior metrics and the external environment. To address these issues, a comprehensive understanding and modeling of user behavior is required.

III. AN AUTONOMOUS FRAMEWORK FOR P2P TOPOLOGY MANAGEMENT

In our topology management framework, we propose to model user behavior to enable the estimation of its activities. Based on the estimation, we propose a stabilization strategy to improve the performance. This work is integrated in a framework that follows IBM's autonomic computing MAPE-K (Monitor, Analyze, Plan, Execute, Knowledge) control loop [14]. In this architecture, the managed element is a hardware or software resource that gets an autonomic behavior by coupling it to an autonomic manager. Autonomic manager monitors

the managed element through data collecting sensors. A plan is prepared through analyzing the collected data and internal knowledge of the system. Finally control actions are executed through effectors.

The instantiation of the MAPE-K loop to our use-case is achieved as follows. We devise a framework that includes monitoring in the form of data collection required to predict the behavior. That data are analyzed through user behavior models and predictions are made. Based on these predictions and the knowledge of the network, a plan is prepared to reduce the impact of peers' departures. Finally, each peer plans to move itself towards the outskirts, before the estimated departure. We discuss these parts of our framework one by one starting with user behavior models that are of two kinds, namely non-contextual and contextual.

A. Understanding user behavior

We have collected large-scale measurements from the literature and analyzed them to extract variables from them which are related to user behavior. We show an abstract view of these measurements in Table I. Measurements over P2P, client/server and IPTV architectures are included to get a system-independent view of user behavior. Data collection methods during these measurements include logs, passive monitoring and active crawlers. As evident from the table, each measurement targets one or a few metrics of user behavior. Moreover, observations concerning relationships among different variables is more sparse and one single study cannot be used to develop a complete model of user behavior. Therefore, we synthesize these measurements to compare and contrast findings about same metrics in different measurements. Moreover, we collect and combine the sparse and split information together to get a complete view of user behavior. We extract from these measurements variables concerning user behavior and their impacting factors. The relationships among all these variables provide a basis for user behavior modeling.

1) *Non-contextual approach:* In this approach [30], we only consider the history of session durations to estimate the current one. Other impacting factors are not taken into account. We propose two estimators in this group. Firstly, we make use of Exponential Moving Average (EMA) that estimates an average by giving exponentially decreasing weights to older sessions. Through this model, the length of the current session ES_t can be estimated from the previous actual session duration S_{t-1} , previous estimated session duration ES_{t-1} and $\alpha \in [0, 1]$ as shown in (1). Here, α is a weighting factor whose higher values give more weights to recent observations.

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

Secondly, we propose a model based on Bayes' rule. This is a probabilistic approach which unlike the previous one, can estimate the current session duration in the presence of no or very few observations. We adapt this approach to our scenario in the following way.

Let $T \in \mathbb{R}$ be the length of longest session of a peer A . We split T in k time steps $\{t_1, t_2 \dots t_k\}$ such that $t_1 < t_2 \dots t_{k-1} < t_k$. The prior probability of $(S_i \geq t_j)$ is given

¹<http://www.cctv.cn/>

| Reference | Type | System | Period | Method | Metrics | | | | |
|------------|---------------------|---------------|----------------------------|----------------------|-------------------------------------|----|----|-----|-----|
| | | | | | OD | SD | CP | A/D | D/U |
| [15] | P2P | PPLive | Feb. to Nov. 2008 | Crawler | ✓ | ✓ | ✓ | | ✓ |
| [5] | | | Apr. to Dec. 2006 | | ✓ | ✓ | | | |
| [10] | | | Nov. 2006 (about 28 hours) | | ✓ | | ✓ | | |
| [6], [7] | | | 2006-07 | Passive/Crawler | ✓ | ✓ | ✓ | ✓ | |
| [8] | | | Jun. 2006 | Passive | ✓ | | | ✓ | |
| [16] | | PPStream | unknown | Crawler | | ✓ | ✓ | ✓ | ✓ |
| [17] | | Zattoo | Mar. 2008 (2 weeks) | | ✓ | | | ✓ | |
| [18] | | CoolStreaming | Mar. 2005 (4 days) | | ✓ | ✓ | ✓ | | |
| [19] | | | (1 day) | | ✓ | ✓ | ✓ | | |
| [20] | | | Oct. 2006 (1 day) | | ✓ | ✓ | ✓ | | |
| [21], [22] | | | Sep. 2006 (1 day) | | ✓ | ✓ | ✓ | ✓ | ✓ |
| [23] | | unknown | 2006 (11 hours) | | | ✓ | | | ✓ |
| [24] | | GridMedia | Jan. 2006 (4.5 hours) | | ✓ | | | | |
| [13] | | Logs | May to Jun. 2008 (5 days) | | ✓ | ✓ | | | ✓ |
| [9] | | | UUSee | | Feb. 2005/Jan. 2006 (2 pop. events) | ✓ | | | |
| [9] | | | CCTV ¹ | Oct. to Jan. 2004-05 | ✓ | | | | |
| [25] | | | Akamai | Oct. to Jan. 2003-04 | ✓ | ✓ | ✓ | | |
| [26] | | | unknown | 2002 (90 days) | ✓ | ✓ | ✓ | ✓ | |
| [27] | | C/S | Telco-Managed | Apr. 2008 (6 days) | ✓ | ✓ | ✓ | ✓ | |
| [28] | Jun. 2008 (1 month) | | | | | ✓ | ✓ | | |
| [29] | May to Oct. 2007 | | | | ✓ | ✓ | ✓ | | |

OD: Online Duration; SD: Session Duration; CP: Channel Popularity; A/D: Arrivals/Departures; D/U: Download/Upload traffic

TABLE I. OVERVIEW OF USER BEHAVIOR MEASUREMENTS

by (2a) where $|O|$ is the number of total already observed sessions. Among them, α_j are the number of sessions for which $(S_i \geq t_j)$. After observing a new set O' of some session durations, the posterior probability of a session duration S_i to be at least equal to t_j is computed through (2b) where $o_j \subseteq O$ and $|o_j|$ is the number of observations where $S_i \geq t_j$.

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

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

Each peer maintains a list of probabilities for each t_j and updates this list at the end of each session after estimating posterior probabilities.

2) *Contextual approach*: In this approach, we consider all the known variables involved in user behavior. We propose a Bayesian network model [31] for user behavior since it suits to such situations where variables contain dependency relationships. We derive our Bayesian network from the results obtained through the synthesis of user behavior measurements [32]. We include all variables into this model that make part of the user behavior or impact them [33], [34]. The resultant Bayesian network model is shown in Figure 1.

We let this network learn its parameters from data. We adopt an incremental learning method, in which a dedicated Bayesian network for each peer learns the behavior. With the passage of time, more observations become available and the network updates its parameters.

B. Planning

Based on estimations of user behavior models, we propose a cooperative strategy for push-based systems to build and maintain a stable tree. In this approach [35], a peer records its joining time after arrival and estimates the length of its current session through using one of the estimators. It then

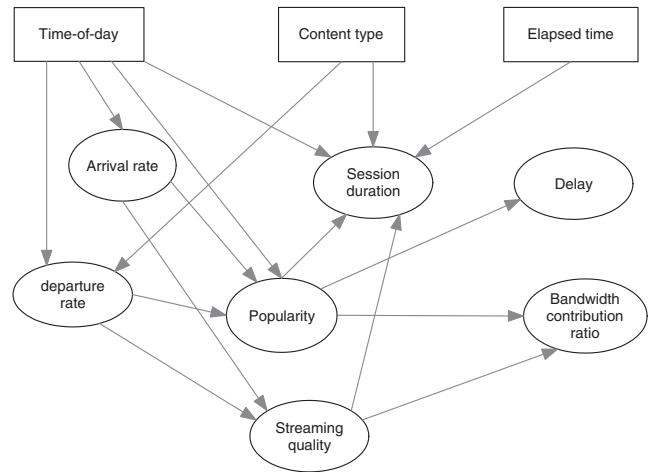


Fig. 1. The Bayesian network model of user behavior

schedules to move itself towards the leaves of the tree before its estimated departure. For such a move, the departing peer attempts to swap its position with the most stable child among its child peers. Otherwise, the most stable child takes its position and the departing peer is moved towards the leaves of the tree. Choosing a stable child and planning a swap or move is outlined in Algorithm 1. Regarding security issues, we assume in this work that peers actually cooperate so that none of them (e.g. free-riders) behave in a malicious way by providing erroneous estimation of their session duration.

A departing peer (B in Figure 2) runs this algorithm to prepare for a swap or move, if it has any child peer otherwise it is already at the leaf and its departure will not impact others. In the first step, B sends a request to all its child peers to get their estimated remaining online time. Each child peer calculates its remaining online time from its estimated session duration and recorded joining time which is sent to the parent peer. This process at the parent peer is outlined in lines 2 to 10.

Algorithm 1 Plan preparation

Declaration:

LRT: Integer ▷ Longest Remaining Time
child, SC: Node ▷ Child Identifier
RT: Integer ▷ Remaining Time
CT: Integer ▷ Current Time
RTQM: Message ▷ Remaining Time Query
RTRM: Message ▷ Remaining Time Response
CQM: Message ▷ Child Query
CRM: Message ▷ Child Response
CPL: List{Node} ▷ Child Peers
RTRL: List{RTRM} ▷ Remaining Time Responses
PPL: List{Node} ▷ Potential Parents

```

1: function STABLENODESELECTION
2:   CPL ← GETCHILDREN
3:   for i ← 1, CPL.size() do
4:     child ← CPL.get(i)
5:     SEND(RTQM, child)
6:   end for
7:   while RTRL.size() < CPL.size() do
8:     RTRM ← RECEIVE(any)
9:     RTRL.add(RTRM)
10:  end while
11:  LRT ← 0
12:  for i ← 1 : RTRL.size() do
13:    RTRM ← RTRL.get(i)
14:    RT ← RTRM.RT
15:    if RT > LRT then
16:      LRT ← RT
17:      SC ← RTRM.source
18:    end if
19:    PPL.add(SC)
20:  end for
21:  SEND(CQM, SC)
22:  CRM ← RECEIVE(SC)
23:  PPL.add(CRM.grandChildren)
24:  return PPL
25: end function

```

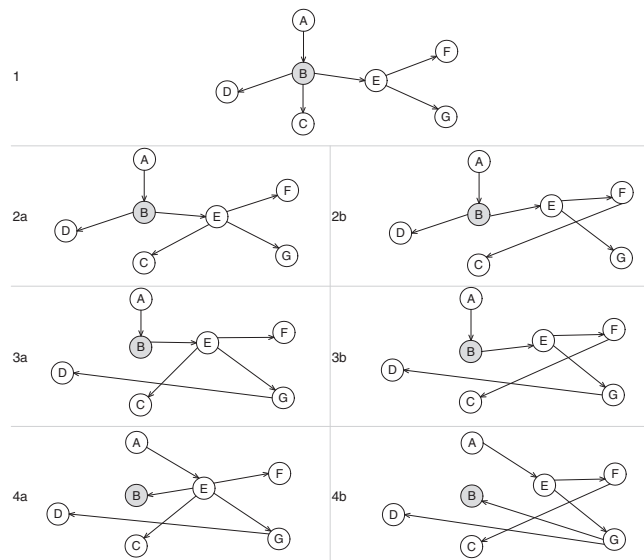


Fig. 2. Moving a departing peer to the leaf

Then, the departing peer B compares all responses to choose the most stable one among them as outlined in lines 11 to 20. Let us suppose this peer is E in our scenario. Firstly, peer B attempts to process a swap with E . To do so, peer A can directly replace peer B by peer E , since it can easily allot the allocated resources for peer B to peer E . However, it is not straightforward for B to become a child of peer E , since peer E may have gone out of resources to accommodate a new child. In that case, B cannot be added as a child peer. Furthermore, other child peers of B (C and D), should no more remain connected to B to avoid stream disruption to them and their descendent peers. Therefore, these child peers need to know about their potential parents to move to them. Peer B has to prepare a list of such peers for itself and its other child peers.

To avoid flooding messages in the network for this purpose, B queries its stable child E for its child peers. Peer E prepares a list of its child peers, including F and G in our case, and sends it to peer B . On receiving this list, peer B prepares a list of potential parents by adding E to it. Lines 21 to 24 of the algorithm outline this process. Now, the plan is ready for execution and peer B will initiate it.

C. Processing changes

Execution of the plan has been outlined in Algorithm 2. Firstly, the remaining child peers of B should move to other peers to avoid stream disruption to them. To do so, the departing peer (B) notifies these peers (C and D) to move to a new parent among the potential ones as demonstrated by lines 2 to 7 of the algorithm. Each of them, iteratively sends a join request to the potential parents, unless the request is accepted or the list is exhausted. In the latter case, it has to perform a random join through the prescribed mechanism of the network. Figures 2 (2a, 2b, 3a and 3b) depict these steps. To accept a child, the only condition is that a peer must have the available capacity. After notifying its other child peers, B sends a replacement request message to its parent peer A (line 9). On receiving this request, peer A replaces child B by E and notifies this to peer E . After receiving the notification, peer E updates its parent peer and checks its available capacity for accepting peer B as a child. In case the capacity is available, it adds peer B as a child, otherwise peer B cannot be added. In either case, peer E has to inform peer B through an acknowledgment message. Figure 2.4a and Figure 2.4b respectively show these two cases.

As soon as, peer B receives the acknowledgment message from peer E , it removes peer E from the list of its own child peers as demonstrated by lines 10 and 11. Moreover, it checks if it has been added by peer E as a child. In case of affirmation, it adds peer E as its parent. Otherwise, it first removes peer E from the list of potential parents and then requests iteratively each peer of the potential parent, unless it is accepted as a child. In case of failure, it performs a random join. Lines 12 to 29 of the algorithm describe this process. It is noticeable from Figure 2.4a and Figure 2.4b that peer B is moved to the leaf in both the cases. Hence, its departure will not disrupt the stream to other peers, and the quality of streaming will consequently improve.

Algorithm 2 Plan execution

Declaration:

accepted: Boolean ▷ Child acceptance
parent: Node ▷ The parent Node
RRM: Message ▷ Replacement Request
RAM: Message ▷ Replacement Acknowledgment
MM: Message ▷ Move Message (to inform a child move to another parent)
JM: Message ▷ Join Message (to subscribe to a parent)
JAM: Message ▷ Join Acknowledgment

```

1: procedure NODESWAP(PPL: List{Node}, SC: Node)
2:   CPL ← GETCHILDREN
3:   for  $i \leftarrow 1, CPL.size()$  do
4:     MM.PPL ← PPL
5:     child ← CPL.get( $i$ )
6:     SEND(MM, child)
7:   end for
8:   parent ← GETPARENT
9:   SEND(RRM, parent)
10:  RAM ← RECEIVE(SC)
11:  CPL.remove(RAM.source)
12:  if SAM.status = false then
13:    accepted ← false
14:    PPL.remove(RAM.source)
15:    for  $i \leftarrow 1, PPL.size()$  do
16:      PP ← PPL.get( $i$ )
17:      SEND(JM, PP)
18:      JAM ← RECEIVE(PP)
19:      if JAM.status = true then
20:        parent ← JAM.source
21:        accepted ← true
22:      end if
23:    end for
24:    if accepted = false then
25:      JOIN
26:    end if
27:  else
28:    parent ← RAM.source
29:  end if
30: end procedure

```

IV. VALIDATION

We have evaluated and compared the accuracy of user behavior models through simulations and the performance improvement through experiments over PlanetLab. To reproduce realistic user behaviors, we used synthetic traces generated through a semi-Markovian process. The synthetic traces generator² [36] is a by-product of our thesis since real traces for individual users are not available due to the difficulty in identifying a user over different appearances. This model is based on the findings presented in the measurement studies research literature. It can generate traces for six different kinds of fictional behaviors called personas. These behaviors are named as Johnatan, Emma, Stephan, Anna, Peter and ELLen. In this document, we abbreviate them as $\{J, E, S, A, P, L\}$ respectively. In order to validate our model, we have compared synthetic traces with models extracted from measurement campaigns. The obtained CDF of a population as well as its evolution over a day are depicted in Figure 3 which exhibits consistent shapes.

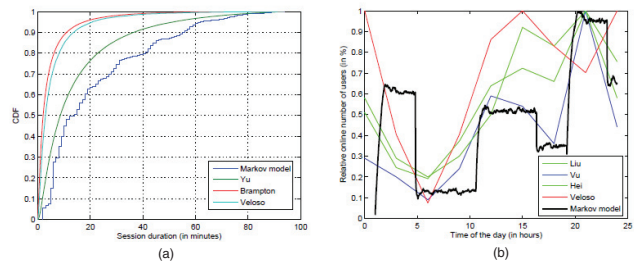
²http://gforge.utt.fr/gf/project/ub_p2p_vs/


Fig. 3. Validation of the trace generator: (a) CDF of session durations ; (b) Population evolution over a day

A. Simulations

We generate traces for 1000 users who join and leave the system during a period of 200 days. Since the Bayesian network requires training, we provide the first 60 days of traces to it as a learning part. We evaluate the Bayesian network model and the non-contextual models over the traces of remaining days. For the Bayesian network model, we use the Bayes Net Toolbox³.

On arrival of each user, his/her session duration is estimated through all the three models. To compare these models for accuracy, we depict the average RMSE over 10 days period with a 95% confidence interval for the users of three representative personas in Figure 4.a.

It is obvious from the figure that the contextual model (BN) produces the least error among all the estimators. The non-contextual models (EMA and BR) not only produce larger errors but also they are not consistent over different personas. These results clearly indicate that contextual approach is more accurate than the non-contextual one.

B. Experiments

We have evaluated the efficiency of our user behavior models and the stabilization strategy proposed in algorithms 1 and 2 through experiments over PlanetLab⁴. In total, we use 60 PlanetLab nodes. We use the open source implementation of a push-based system, namely Scribe⁵. In total we perform four distinct experiments. One of them uses Scribe in its original form without involving any stabilization strategy. We denote this experiment through WS. The three other experiments integrate the implementation of stabilization strategy. Among them, each experiment uses a different estimation technique. We name them through the name of estimators utilized: BN, EMA and BR.

We choose to replay the two hours traces from the data we already used for simulations earlier. These traces consist of first two hours of day 61. Likewise simulations, we train our Bayesian network over traces of first 60 days. The experiment starts from the first minute of day 61. Among the 1000 considered users, each of them joins the system as its joining time matches with the current time. This peer remains online for a period of time equal to the length of its actual session.

³<http://code.google.com/p/bnt/>
⁴www.planet-lab.org
⁵<http://www.freepastry.org/>

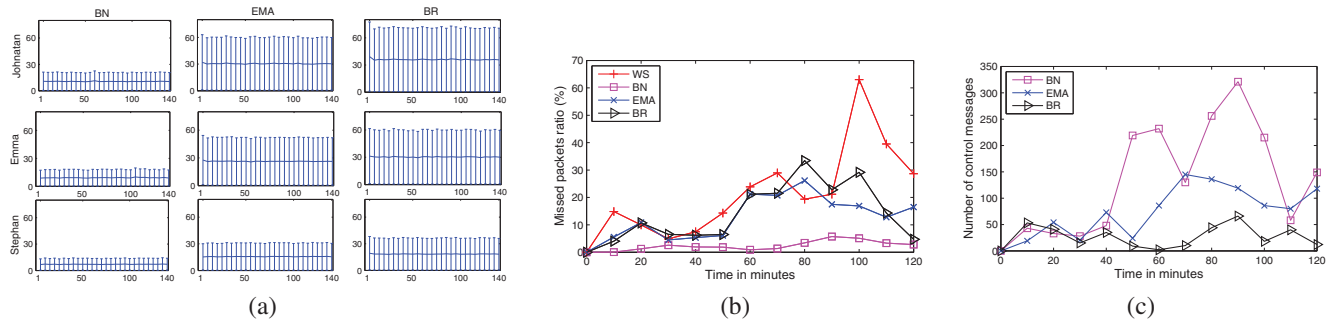


Fig. 4. (a) Averaged RMSE over a 10 days period (x-axis: number of days, y-axis: averaged RMSE); (b) Lost packets during each experiment; (c) Communication cost of each approach

The estimated length of its session through the given estimator is used by the stabilization algorithms. One peer is fixed to remain online throughout the experiment and it serves as a streaming source. The Streaming bit rate is set to 64 Kbps. A peer can accommodate a maximum of three child peers.

During each experiment, we analyze the impact of peers departures and improvement in performance through our approaches. In particular, we measure missed packets, buffering delay, initial streaming quality, played packets, average delay, and the communication overhead of the stabilization strategy. Due to space constraints, we present here results concerning missed packets and the communication overhead of our approach.

Missed packets are those which are lost or arrive at a peer after the playback deadline is over. We depict the ratio of missed packets to played ones (those which arrived before the playback deadline) in Figure 4.b. An average of all such packets over each 10 minutes has been taken for demonstration purpose. It is clear, that Scribe without stabilization (WS) has the largest ratio at most of the instances among all the experiments. At the other hand, the stabilization strategy relying on the contextual approach (BN) shows minimum packet losses than other approaches. Overall, this approach reduces the packet losses of Scribe by 89%, which is the highest among all. Among non-contextual approaches, EMA reduces the losses by 38% while BR by 30%.

The communication overhead of the strategy lies in the number of control messages exchanged for stabilization. We measure the number of these messages and depict them in Figure 4.c. The total number of control messages during each 10 minutes interval for each experiment are grouped together. Here, the stabilization strategy relying on BN incurs a larger overhead than other estimators, however, an average of 14.4 messages are being sent by all peers in the system during one minute time, which is a negligible amount of messages.

Concerning other performance metrics such as delay and streaming quality, BN produces comparable results to other estimators. In light of these results the contextual model outperforms the non-contextual ones due to its improved accuracy by considering all the contextual information of user behavior.

V. CONCLUSION AND FUTURE PERSPECTIVES

In this dissertation, we focused on modeling and integration of user behavior into P2P live streaming systems. User behav-

ior has a deep impact on the performance of these systems, since peers are not fixed but behave as their users do. To mitigate the impact of their intermittent presence, currently deployed systems build resilient structures which lead to QoS and QoE issues.

To overcome these issues, we focused directly on user behavior. We performed an extensive synthesis of user behavior measurements in video steaming systems to get a thorough understanding of it. In light of this synthesis, we proposed two kinds of models, namely, non-contextual and contextual. The former approach ignores any contextual information while the latter considers all the contextual variables that can impact the behavior. We also proposed a stabilization strategy for push-based systems, that relies on the estimations of behavior models to know the stability of peers and moves the unstable ones towards the leaves of the tree.

We performed simulations and experimentations over Plan-etLab to evaluate these approaches. Results show that the contextual model estimates with better accuracy than the non-contextual ones. Moreover, the stabilization strategy improves the performance of an existing system significantly. In particular, coupled with the contextual model the improvement is superior to others.

This work opens several future perspectives. Short term ones are related to its use in extended or different control algorithms and its application to others streaming models, while long term perspectives consist in its use for other kind of services. Firstly, the contextual model can be utilized in an extended version of the stabilization strategy that could include bandwidth contribution ratio and streaming quality of a peer along with its stability to decide its location in the overlay. Also, beyond single channel systems, models of user behavior can be applied to cross channel systems to predict the next channel a user will most probably switch to. Choosing such channels for relaying to other peers earlier can minimize the switching delay considerably. Secondly, this work could also be envisaged for other management purposes. For example, a service provider could use the Bayesian model to get an estimation of the population at a given time and deploy resources accordingly, helping thus for capacity planning issues. Finally, this work is a first step toward the integration of the user behavior into management mechanisms and its results are promising. It appears as an important ground block toward the design of user-aware services in the future Internet.

VI. THESIS MATERIAL

Thesis manuscript and the open source software for traces generation are available at http://gforge.utt.fr/gf/project/ub_p2p_vs/. Thesis international publications are listed in the references section, numbered as [30], [31], [32], [33], [34], [35], [36].

REFERENCES

- [1] Y. Chu, S. G. Rao, S. Seshan, and H. Zhang, "A case for end system multicast," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 8, pp. 1456–1471, 2002.
- [2] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum, "CoolStreaming/DONet: A data-driven overlay network for Peer-to-Peer live media streaming," in *IEEE INFOCOM*, 2005, pp. 2102–2111.
- [3] F. Wang, Y. Xiong, and J. Liu, "mTreebone: A collaborative tree-mesh overlay network for multicast video streaming," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, pp. 379–392, 2010.
- [4] Y. Liu, Y. Guo, and C. Liang, "A survey on Peer-to-Peer video streaming systems," *Peer-to-Peer Networking and Applications*, vol. 1, no. 1, pp. 18–28, 2008.
- [5] L. Vu, I. Gupta, J. Liang, and K. Nahrstedt, "Measurement and modeling of a large-scale overlay for multimedia streaming," in *The Fourth International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness & Workshops (QSHINE)*, 2007, pp. 1–7.
- [6] X. Hei, C. Liang, J. Liang, Y. Liu, and K. Ross, "Insights into PPLive: A measurement study of a large-scale P2P IPTV system," in *IPTV Workshop*, 2006.
- [7] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, "A measurement study of a large-scale P2P IPTV system," *IEEE Transactions on Multimedia*, vol. 9, no. 8, pp. 1672–1687, 2007.
- [8] T. Silverston, O. Fourmaux, A. Botta, A. Dainotti, A. Pescapè, G. Ventre, and K. Salamatián, "Traffic analysis of Peer-to-Peer IPTV communities," *Computer Networks*, vol. 53, no. 4, pp. 470–484, 2009.
- [9] Y. Tang, L. Sun, J.-G. Luo, and Y. Zhong, "Characterizing user behavior to improve quality of streaming service over P2P networks," in *Advances in Multimedia Information Processing (PCM)*, 2006, pp. 175–184.
- [10] F. Wang, J. Liu, and Y. Xiong, "Stable peers: Existence, importance, and application in Peer-to-Peer live video streaming," in *IEEE INFOCOM*, 2008, pp. 1364–1372.
- [11] B. Liu, Y. Cao, Y. Cui, Y. Xue, F. Qiu, and Y. Lu, "Minimizing service disruption in peer-to-peer streaming," in *IEEE CCNC*, 2011, pp. 1066–1071.
- [12] Q. Zheng, Y. Long, T. Qin, and L. Yang, "Lifetime characteristics measurement of a P2P streaming system: Focusing on snapshots of the overlay," in *WCICA*, 2011, pp. 805–810.
- [13] Z. Liu, C. Wu, B. Li, and S. Zhao, "Distilling superior peers in large-scale P2P streaming systems," in *IEEE INFOCOM*, 2009, pp. 82–90.
- [14] IBM, "An architectural blueprint for autonomic computing," White Paper, 2006.
- [15] S. Spoto, R. Gaeta, M. Grangetto, and M. Sereno, "Analysis of PPLive through active and passive measurements," in *IEEE International Symposium on Parallel & Distributed Processing*, 2009, pp. 1–7.
- [16] J. Jia, C. Li, and C. Chen, "Characterizing PPStream across Internet," in *IFIP International Conference on Network and Parallel Computing Workshops*, 2007, pp. 413–418.
- [17] K. Shami, D. Magoni, H. Chang, W. Wang, and S. Jamin, "Impacts of peer characteristics on P2PTV networks scalability," in *IEEE INFOCOM*, 2009, pp. 2736–2740.
- [18] X. Zhang, J. Liu, and 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.
- [19] B. Li, Gabriel Y. Keung, Susu Xie, F. Liu, YeSun, and Hao, "An empirical study of flash crowd dynamics in a P2P-based live video streaming system," in *IEEE GLOBECOM 2008*, 2008, pp. 1–5.
- [20] S. Xie, G. Y. Keung, and B. Li, "A measurement of a large-scale Peer-to-Peer live video streaming system," in *International Conference on Parallel Processing Workshops (ICPP)*, 2007, pp. 57–62.
- [21] B. Li, S. Xie, Y. Qu, G. Y. Keung, C. Lin, J. Liu, and X. Zhang, "Inside the new Coolstreaming: Principles, measurements and performance implications," in *IEEE INFOCOM*, 2008, pp. 1031–1039.
- [22] B. Li, S. Xie, G. Y. Keung, J. Liu, I. Stoica, H. Zhang, and X. Zhang, "An empirical study of the Coolstreaming+ system," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 9, pp. 1627–1639, 2007.
- [23] S. Agarwal, "A case study of large scale P2P video multicast," in *International Conference on IP Multimedia Subsystem Architecture and Applications (IMSAA)*. IEEE, 2007, pp. 1–5.
- [24] Y. Tang, L. Sun, K. Zhang, S.-Q. Yang, and 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 (ICME)*, 2007, pp. 2158–2161.
- [25] K. Sripanidkulchai, B. Maggs, and H. Zhang, "An analysis of live streaming workloads on the Internet," in *Internet Measurement Conference (IMC)*. ACM, 2004, pp. 41–54.
- [26] E. Veloso, V. Almeida, W. J. Meira, A. Bestavros, and S. Jin, "A hierarchical characterization of a live streaming media workload," *IEEE/ACM Transactions on Networking*, vol. 14, no. 1, pp. 133–146, 2006.
- [27] T. Qiu, Z. Ge, S. Lee, J. Wang, J. Xu, and Q. Zhao, "Modeling user activities in a large IPTV system," in *Internet Measurement Conference (IMC)*. ACM, 2009, pp. 430–441.
- [28] T. Qiu, Z. Ge, S. Lee, J. Wang, Q. Zhao, and J. Xu, "Modeling channel popularity dynamics in a large IPTV system," in *11th international joint conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, 2009, pp. 275–286.
- [29] M. Cha, P. Rodriguez, J. Crowcroft, S. Moon, and X. Amatriain, "Watching television over an IP network," in *Internet Measurement Conference (IMC)*. ACM, 2008, pp. 71–84.
- [30] I. Ullah, G. Bonnet, G. Doyen, and D. Gaïti, "Improving performance of ALM systems with Bayesian estimation of peers dynamics," in *12th IFIP/IEEE International Conference on Management of Multimedia and Mobile Networks and Services (MMNS)*, 2009, pp. 157–169.
- [31] —, "Modeling user behavior in P2P live video streaming systems through a Bayesian network," in *IFIP International Conference on Autonomous Infrastructure Management and Security (AIMS)*, 2010, pp. 2–13.
- [32] I. Ullah, G. Doyen, G. Bonnet, and D. Gaïti, "A survey and synthesis of user behavior measurements in video streaming systems," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 3, pp. 734–749, 2012.
- [33] —, "User behavior anticipation in P2P live video streaming systems through a Bayesian network," in *12th IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2011, pp. 337–344.
- [34] —, "A bayesian approach for user aware peer-to-peer video streaming systems," *Elsevier Journal of Signal Processing: Image Communication*, vol. 27, no. 5, pp. 438–456, 2012.
- [35] —, "An autonomous topology management framework for QoS enabled P2P video streaming systems," in *International Conference on Network and Service Management (CNSM)*. IEEE, 2012, pp. 126–134.
- [36] G. Bonnet, I. Ullah, G. Doyen, L. Fillatre, D. Gaïti, and I. Nikiforov, "A Semi-Markovian individual model of users for P2P video streaming applications," in *4th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 2011, pp. 1–5.