Modeling Interactive Digital TV Users Behavior

Samuel da Costa Alves Basilio

Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Leopoldina - MG - Brazil
Email: samuelbasilio@leopoldina.cefetmg.br

***Abstract*—The performance evaluation required in the proposition of new large scale distributed systems usually faces the challenge of correct characterization of the load that is imposed on them. In the case of proposals in the area of Digital Television (TV), including terrestrial, cable, Satellite and IPTV, obtaining such a characterization from real deployment scenarios has proved to be a very difficult accomplishment due to the impediment experimental access to these distribution networks in operation. Thus, a researcher usually uses simulations that impose work loads crudely approximated, or even fictitious oversized to your system, leading uncertainty to potential service providers regarding the optimal sizing of the equipment required. Here, we present a mathematical model of simple implementation, able to represent the behavior of users of Digital TV. The model can be parameterized to represent different states of behavior about the system to be simulated, and thus adapt to various scenarios of interest. We also show how this model was used in the performance analysis of a proposed service provider.**

***Keywords*–Digital TV; Model; User Behavior; Interaction; Simulation.**

I. Introduction

When dealing with research in Digital TV, we often face a challenge in time to validate the software developed. As this environment involves millions of users, the developed software have to take into account concepts such as scalability, availability and performance, but the effort to evaluate these features may not be trivial. Undoubtedly, the best way to improve, correct errors and check the software requirements is applying it in a real environment, even if with a restricted group of users. In the case of Digital TV environment, this done has proven to be difficult due to the fact that the real environment is also commercial. Additionally, experimental tests are not generally accepted. Although the use of a limited number of users is not ideal, in most cases, it is the resource available for researchers and software developers. This feature can be presented as a good solution when we want to evaluate interface, functionality, among others factors. However, when we want to evaluate such criteria as scalability and availability, it is not enough. In this case, a resource that can be used efficiently, cheaply and reliably is simulations. Nowadays, computational resources are relatively inexpensive and can be used to simulate environments with a large number of users. However, to use this resource, we need a reliable model, which should represent as closely as possible the environment behavior. The challange to use this resource is in the development and implementation of the model to be simulated. It is necessary that the model developer closely observes the behavior of all the environment elements and abstract then them in a simple model. The model must have a balance between fidelity to the real environment and simplicity of implementation. Moreover, a behavior model must suit to its purpose. We should not use a desktop user behavior model to simulate a mobile phone environment. Similarly, we should not use a web user behavior model to simulate an interactive digital TV load.

Knowing these problems, we present in this paper a mathematical model that can easily be implemented and even so is faithful enough to reality. Furthermore, with the data from an experiment where 27 viewers had their interactions with TV capture along with a TV audience survey data from a local statistics research institute, we show a parameterization example of the proposed model. The presented model may also be used in other contexts, such as targeted advertising and social context analysis, audience measurement, among others.

This paper is structured as follows. Section II discusses some related work; Section III is dedicated to mathematical model of the behavior of users of Interactive Digital TV. Section IV exemplifies the model instantiation for a specific case. Finally, Section V shows the achieved objectives and discusses future work.

II. Related Work

The attempt to model the behavior of media consumption system users is not a new work as seen in [1], where Branch et al. characterize and model the behavior of users of their video on demand system. But, as new ways of interaction appear, as well as new technologies and new systems of media consumption, existing models are often not adequate. Alvarez et al. [2] shows an architecture for audience measurement, a model for data consumption and some metrics to quantify the impact of consumer. This metric is calculated in a similar way in [3].

Along with the user behavior model, some work show a characterization of this behavior in a real environment. That is the case of [4], which characterizes the behavior from a system with millions of users. An interesting metric presented in this paper is the session time that has a paramount importance when simulating the behavior of various users over long periods of time.

An important point is cited in [5], where a synthetic load generator is shown. In this paper, Costa et al. cite the need for heterogeneity in load generators because many are reported in the literature, but most work only a group-specific data, such as educational. The work presented by Qiu et al. [6] also has a generator of synthetic load but only focusing on the Internet Protocol Television (IPTV) environment. Nevertheless, this work has an advantage because it used data from a real system with millions of users.

 ISBN: 978-1-61208-416-9

III. Interactive Digital TV Users Behavior Model

Aiming to support possible future simulations, we present in this session a Markov model of the Interactive Digital TV users interactions. This work comes to fill a gap in the digital TV research. It is really hard to evaluate points like scalability, availability and performance at this area because until now there is no model to represent the interactive digital TV user behavior.

To define the Interactive Digital TV user behavior, we must consider every possibility of viewer interaction. Below, we define the possible states its transitions. Figure 1 shows the states and the possible transitions between these states:

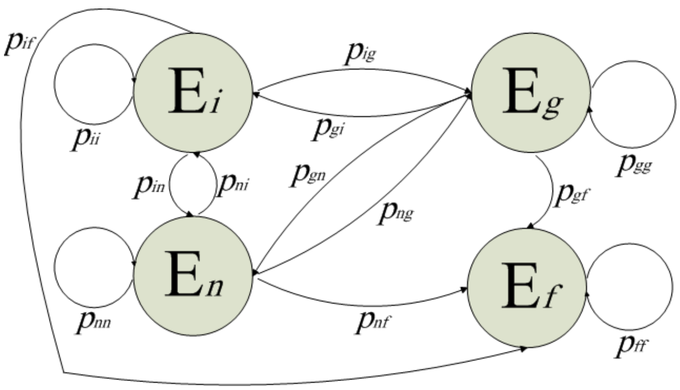


Figure 1. Markovian model of the users interactions.

- E_i, when an interactive application is running;
- E_n, when only the broadcast video is running;
- E_f, when the viewer turns off the digital receiver;
- E_g, when a native application is running.

- p_{if}, probability of the digital receiver to be unplugged since an interactive application was running;
- p_{ii}, probability of the viewer to continue the execution of an interactive application;
- p_{ig}, probability of the viewer start a native application as an interactive application was running (pausing the execution of the interactive application);
- p_{in}, probability of a running interactive application to be closed or the viewer change the channel (ending the execution of the application);
- p_{ni}, probability of an interactive application to be started since no other application was running;
- p_{nn}, probability of the viewer does not start any application;
- p_{nf}, probability of the receiver to be turned off without any application be running;
- p_{ng}, probability of the viewer start a native application;
- p_{gn}, probability of the viewer select a channel through a native application (such as the electronic program guide);
- p_{gg}, probability of the viewer to continue running a native application;
- p_{gf}, probability of the receiver be turned off with a native application running;
- p_{gi}, probability of a native application to be terminated after being initiated when an interactive application was running (recovering the state of the interactive application that was then paused);
- p_{ff}, probability of the receiver be off.

We differentiate the state where native applications (we consider native applications those that are specific to the digital receiver, from the factory or installed later as the electronic program guide.) are running from the state where interactive applications in general are because depending on data capturing approach used it may not be possible to capture the interactions of native applications [7]. As these native applications are a resource provided by the digital receiver, if the data capture approach by interactive applications is used, user interaction with these native applications can not be obtained. However, if the data capture approach used is by middleware extension, all interactions may be obtained, the ones with interactive applications such as the ones with native applications. In both approaches, the channel change interactions can be obtained.

Note that it is simple to specialize the presented model to any specific case. For example, if we wanted to specify at the initial model an arbitrary interactive application, it would only need to add to each state E_x of the application the probabilities p_{xy} and p_{yx}, where m is the number of states of the chain $E_x = \{E_{a1}, E_{a2}, ..., E_{am}\}$, $p_{xy} = \{p_{xn}, p_{xg}, p_{xi}, p_{xf}\}$ and $p_{yx} = \{p_{nx}, p_{gx}, p_{ix}, p_{fx}\}$. We also have to remove the state E_f, which represents the final state of the application chain, as this state is reached at some point when any of the probabilities p_{xy} happen.

To illustrate the specialization of the model Figure 2(a) shows the model of an arbitrary interactive application that has two active states, E_{a1} and E_{a2}, and a final state E_f. To extend the original model we remove the state E_f from the application model and add the states E_{a1} e E_{a2} at the original model, keeping the probabilities that relate the states E_{a1} and E_{a2} and adding the probabilities that relate E_{a1} and E_{a2} with $E_i^{'}$, E_g, E_n e E_f. Note that the estate E_i that at the original model represented all interactive application, is now called $E_i^{'}$ and represente all others interactive applications. If all applications are represented individually, the state $E_i^{'}$ can be removed from the model. The same reasoning can be applied at the state E_g. Figure 2(b) shows the final result.

A. *Expected Session Time*

An interesting metric that we can obtain from this model is the expected session time. The session time of a viewer can be estimated by calculating the number of steps required to reach the state E_f. Below is the formalism to calculate this time, taken from [8].

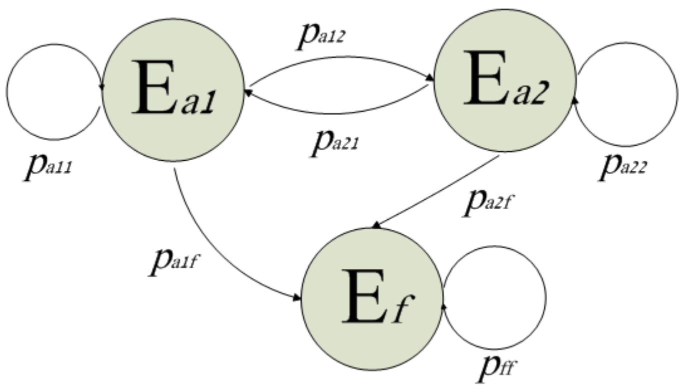
Let $(X_n)_{n\geq 0}$ be a Markov chain with transition matrix P. The *hitting time* of a subset A of I is the random variable $H^A : \Omega \to \{0, 1, 2, ...\} \cup \{\infty\}$ given by:

$$H^A(\omega) = inf\{n \geq 0 : X_n(\omega) \in A\}$$

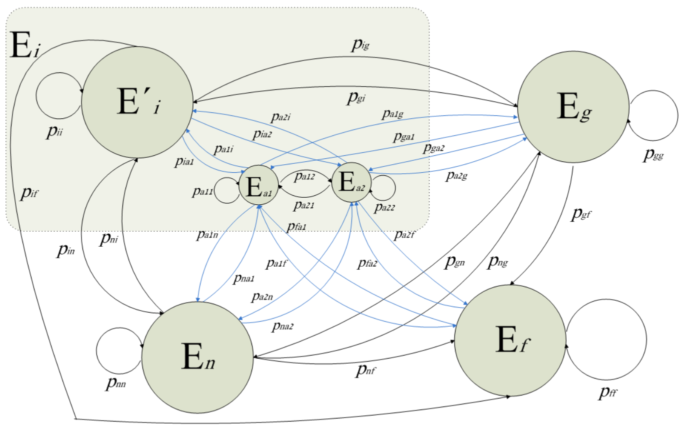
where the infimum of the empty set $\emptyset$ is ∞. The probability starting from i that $(X_n)_{n\geq 0}$ ever hits A is then:

$$h_i^A = P_i(H^A < \infty).$$

 ISBN: 978-1-61208-416-9



(a) Model of an arbitrary application



(b) Specialized model

Figure 2. Model specialization.

When A is a close class, h_i^A is called the *absorption probability*. The mean time taken for $(X_n)_{n\geq 0}$ to reach A is given by:

$$k_i^A = E_i(H^A) = \sum_{n<\infty} nP(H^A = n) + \infty P(H^A = \infty).$$

For our case, we have $i = \{E_n\}$ and $A = \{E_f\}$. It is clear that $k_{E_f} = 0$. Starting at E_n and performing one step, with probability p_{ni} we reach E_i, with probability p_{ng}, we reach E_g, with probability p_{nf} we reach E_f, and with probability p_{nn} we keep at E_n. Then:

$$k_{E_n} = 1 + p_{ni}k_{E_i} + p_{ng}k_{E_g} + p_{nf}k_{E_f} + p_{nn}k_{E_n}$$
$$k_{E_n} = \frac{1 + p_{ni}k_{E_i} + p_{ng}k_{E_g}}{(1 - p_{nn})}$$

The 1 appears because we count the time for the first step. Similarly, for k_{E_g} and k_{E_i}, we have the following system that solved gives the expected session time:

$$\begin{cases} k_{E_n} = \dfrac{1 + p_{ni}k_{E_i} + p_{ng}k_{E_g}}{(1 - p_{nn})} & \text{(1a)} \\ k_{E_g} = \dfrac{1 + p_{gi}k_{E_i} + p_{gn}k_{E_n}}{(1 - p_{gg})} & \text{(1b)} \\ k_{E_i} = \dfrac{1 + p_{ig}k_{E_g} + p_{in}k_{E_n}}{(1 - p_{ii})} & \text{(1c)} \end{cases}$$

B. Expected Number of Interactions in a Given Period

It is important to note that the probabilities p_{ii}, p_{gg} and p_{nn}, represent the probability of the viewer continue in its current state, running an interactive application, a native application or not running any applications, respectively. If at some step in the execution of this model any of these probabilities happen, an event of interaction can be generated or not. If at the state E_i the probability p_{ii} happen, the viewer continues with the interactive application running. In the time period covered by this step, he may or may not have interacted with the application. There is, then, at this point, a probability of generating a user interaction event. We call this probability *interaction rate at the state*, being noted as pe_{ii}. The same reasoning can be applied to p_{gg} and p_{nn}, and we call pe_{gg} and pe_{nn} their respective interaction rates at the state.

With that, another important metric that we can calculate is the expected number of interactions that a viewer shall perform in a given period. Consider that whenever there is a transition between the states E_i, E_g and E_n an event I is generated. This event is an interaction. Also, when a transition where there is no state change occurs p_{ii}, p_{gg} and p_{nn} an event is generated with probability pe_{ii}, pe_{gg} and pe_{nn} respectively. Thus, the probability that for each state E_i, E_g and E_n, we generate an interaction event is:

$$I = \begin{cases} p_{ni} + p_{ng} + p_{nn}pe_{nn} & \text{, if the current state is } E_n \\ p_{in} + p_{ig} + p_{ii}pe_{ii} & \text{, if the current state is } E_i \\ p_{gi} + p_{gn} + p_{gg}pe_{gg} & \text{, if the current state is } E_g \end{cases} \quad (2)$$

Observing function (2), we note that the probability of generate an event is dependent of the current state. Therefore, owe calculate for each step k the probability of being in each state E_i, E_g and E_n. Let $p_{E_i}^{(k)}$, $p_{E_n}^{(k)}$ and $p_{E_g}^{(k)}$ be the probabilities in step k of being at state E_i, E_n and E_g, respectively, these probabilities are given by the recursive functions system:

$$\begin{cases} p_{E_i}^{(k)} = p_{gi}p_{E_g}^{(k-1)} + p_{ni}p_{E_n}^{(k-1)} + p_{ii}p_{E_i}^{(k-1)} \\ p_{E_n}^{(k)} = p_{gn}p_{E_g}^{(k-1)} + p_{nn}p_{E_n}^{(k-1)} + p_{in}p_{E_i}^{(k-1)} \\ p_{E_g}^{(k)} = p_{gg}p_{E_g}^{(k-1)} + p_{ng}p_{E_n}^{(k-1)} + p_{ig}p_{E_i}^{(k-1)} \end{cases} \quad (3)$$

This way, to calculate the total amount of generated events $I^{(k)}$ in k steps, we must sum for each step k the probabilies of generate an event in each state. Formally:

$$I^{(k)} = \sum_{j=1}^{k} p_{E_n}^{(j)} I + p_{E_i}^{(j)} I + p_{E_g}^{(j)} I \quad (4)$$

C. Expected Number of Applications Execution

We also can estimate the expected number of applications execution. That metric is represented by the expected number of visits to a state. If we extend our model so that each interactive application is represented by an individual state, we even shall be able to estimate the expected execution number of an specific application.

Let,

$$N_j(k) = \sum_{m=1}^{k} I(X_m = j) \quad (5)$$

 ISBN: 978-1-61208-416-9

the number of visits to state j during times 1 to k. And let,

$$G_{ij}(k) = E(N_j(k)|X_0 = i) = \sum_{m=1}^{k} P_{ij}^{(m)} \qquad (6)$$

the average time of visits to state j during times 1 to k, starting at i. If we consider j as our state E_i and i as our states E_n and E_g, the expected number of interactive applications execution in k time is given by

$$G_{E_nE_i} + G_{E_gE_i} \qquad (7)$$

IV. Model Instantiation

As part of the work [7], an experiment was conducted in which 27 volunteers watched television and had their interactions captured. All analyzed viewers watched the available programming for 15 minutes. Using these data we conducted an instantiation of the presented model in order to illustrate its use.

Table I shows the quantification of used data and Table II the probabilities found. In Table I we consider as bounce rate the percentage between the total of executions and total executions where after start the application, the user closed it without making any other interaction.

TABLE I. SUMMARY OF USED DATA.

USED DATA	VALUE
INTERACTIONS TOTAL	2241
CHANNEL CHANGE TOTAL	355
INTERACTIONS WITH INTERACTIVE APPLICATIONS	1886
TOTAL EXECUTIONS OF INTERACTIVE APPLICATIONS	139
AVERAGE TIME OF APPLICATIONS EXECUTION	45,02 secs
BOUNCE RATE	45.32%
BIGGEST NUMBER OF INTERACTIONS OF A SINGLE USER IN A MINUTE	57
USERS TOTAL	27
EXPERIMENT TOTAL TIME	6 hours and 45 mins

For definition of these data, the experiment time was discretized in seconds. From this we calculated the number of seconds that a viewer was in each state E_i, E_g and E_n. Knowing how long each viewer spent in each state and transitions between the states it was possible to get the data from Table II. In the experiment, every session was started in E_n state.

With the probabilities it is possible to calculate the expected session time replacing the obtained values in equations (1a), (1b) e (1c).

$$\begin{cases} k_{E_n} = \dfrac{1 + \frac{118}{13895}k_{E_i} + \frac{35}{13895}k_{E_g}}{(1 - \frac{13728}{13895})} & \text{(8a)} \\ k_{E_g} = \dfrac{1 + \frac{18}{2428}k_{E_i} + \frac{13}{2428}k_{E_n}}{(1 - \frac{2391}{2428})} & \text{(8b)} \\ k_{E_i} = \dfrac{1 + \frac{2}{3477}k_{E_g} + \frac{133}{3477}k_{E_n}}{(1 - \frac{3340}{3477})} & \text{(8c)} \end{cases}$$

Solving the system we find:

$$k_{E_n} = \frac{19903979}{22128} \approx 899.49290 \qquad (9)$$

TABLE II. VALUES FOR THE PROBABILITIES OF OUR MODEL.

PROBABILITY	VALUE
p_{ii}	$\frac{3340}{3477}$
p_{in}	$\frac{133}{3477}$
p_{ig}	$\frac{2}{3477}$
p_{if}	$\frac{2}{3477}$
p_{gi}	$\frac{18}{2428}$
p_{gg}	$\frac{2391}{2428}$
p_{gn}	$\frac{13}{2428}$
p_{gf}	$\frac{6}{2428}$
p_{nn}	$\frac{13728}{13895}$
p_{ni}	$\frac{118}{13895}$
p_{ng}	$\frac{35}{13895}$
p_{nf}	$\frac{14}{13895}$
pe_{nn}	$\frac{1036}{13895}$
pe_{ii}	$\frac{1126}{3477}$
pe_{gg}	$\frac{482}{2428}$

We are also able to calculate the expected number of interactions in 15 minutes. As every session started in E_n state, we have that $p_{E_i}^{(1)} = p_{E_g}^{(1)} = p_{E_f}^{(1)} = 0$ and $p_{E_n}^{(1)} = 1$. Calculating the probabilities for each step k of the system (3) and replacing Table II probabilities in the sum (4), we have:

$$\begin{aligned} I^{(900)} = \sum_{j=1}^{900} & p_{E_n}^{(j)}\left(\frac{118}{13895} + \frac{35}{13895} + \frac{13728}{13895}\frac{1036}{13895}\right) + \\ & p_{E_i}^{(j)}\left(\frac{133}{3477} + \frac{2}{3477} + \frac{3340}{3477}\frac{1126}{3477}\right) + \\ & p_{E_g}^{(j)}\left(\frac{18}{2428} + \frac{13}{2428} + \frac{2391}{2428}\frac{482}{2428}\right) \\ I^{(900)} \approx 82.8720 & \end{aligned} \qquad (10)$$

We will also estimate the amount of interactive applications executions using (7), (3) and data from Table II. For this case, we have that

$$G_{E_nE_i}(k) = \sum_{m=1}^{k} P_{E_nE_i}^{(m)} = \sum_{m=1}^{k} p_{E_n}^{(m-1)} p_{ni} \approx 3.45$$

$$G_{E_gE_i}(k) = \sum_{m=1}^{k} P_{E_gE_i}^{(m)} = \sum_{m=1}^{k} p_{E_g}^{(m-1)} p_{gi} \approx 0.51$$

so:

$$G_{E_nE_i} + G_{E_gE_i} \approx 3.96 \qquad (11)$$

Observing the results (9) and (10), we verified the accuracy of our calculations and probabilities, as the experiment lasted

 ISBN: 978-1-61208-416-9

15 minutes (900 seconds), this was the average time and soon, the expected session time. Also the interactions total is 2241 and the users total 27, as seen at Table I, so the expected number of interactions is 83 interactions per user with a 15 minutes session. (11) shows a larger error. As the total executions of interactive applications is 139 and the users total 27, the expected amount of interactive applications executions per user should be closer to 5. This larger error happens because we ignore the chance of an application to be closed and opened at an interval shorter than 1 second, this way the current state was E_i and has not changed. For simplicity, this possibility is not considered in our calculations.

A. Model Use to Generate Synthetic Load

Also as part of the work presented in [7], we conducted a load test on a real server which implemented an audience and interaction analysis service provider. This service provider is constantly receiving data captured at the viewers digital receiver and store this data in a relational database. The purpose of this load test was to illustrate the use of the presented model. In this case to generate synthetic load for scalability and performance testing. The implementation of this server was made using a virtual machine with the settings shown in Table III.

TABLE III. SEVER CONFIGURATION.

SYSTEM PART	TECHNICAL DETAILS
CPU	Intel Xeon 2.0GHz
RAM	1GB
Cores	1 or 2
Swap	2GB
Operating System	Ubuntu 12.04 LTS
HD	14GB
Web Server	Apache Tomcat/6.0.35
Data Base	MySQL: 5.5.31

In this test, we used the probabilities presented in Table II. We have further increase the probabilities values pe_{ii}, pe_{gg} and pe_{nn} to 1. This way, we assure that every second an interaction would be generated and sent to our server. For the test we send 10000 requests to the server at each session. We started sending batches of 100 simultaneous requests and check the response time. We have been increasing the number of simultaneous requests to reach 500. To generate this amount of requests it takes about 150 to 800 model instances. We did this experiment twice using the same server, at the first time with a single core and at the second time with two cores. Figure 3 shows the result of this test.

As was to be expected, as we increase the number of simultaneous requests the response time for each request also increases. When we send more than 500 simultaneous requests our server starts to fail due to overload. It is also noteworthy that with a small increase in computational power, response time greatly improves on average 3.008 ms.

V. CONCLUSION

In this paper, we present a mathematical model of Interactive Digital TV user interactions. Despite being simple and easy to implement, this model is sufficiently faithful to reality and can be used for the most diverse simulations purposes. We also showed how it is possible to extend the model to more specific and detailed models.

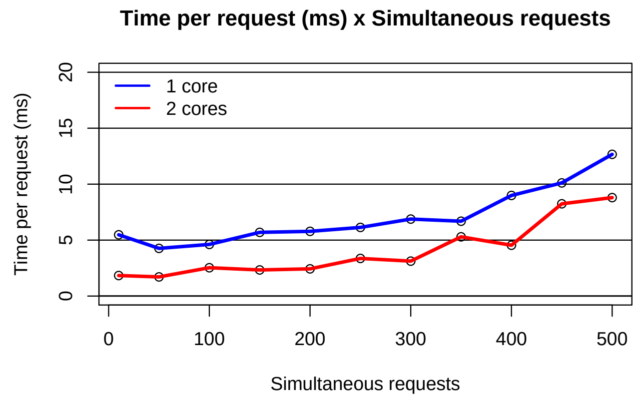


Figure 3. Time taken to complete each request.

We still use the data obtained in a field experiment as input to the model. Thus, we exemplify the use of the model for the calculation of certain metrics and generating synthetic load.

As future work, we propose new extensions to the proposed model, increasing its specialization and complexity. Using data from validity statistical captures would also be very interesting, because with this, we could have a much more reliable numerical model. But, capturing these data is impossible considering the current audience and interactivity measurement approaches.

ACKNOWLEDGMENTS

The authors thank CEFET-MG (Brazil) for financial support for this work.

REFERENCES

[1] P. Branch, G. Egan, and B. Tonkin, “Modeling interactive behaviour of a video based multimedia system,” in Communications, 1999. ICC ’99. 1999 IEEE International Conference on, vol. 2, 1999, pp. 978–982.

[2] F. Alvarez, C. Martin, D. Alliez, P. Roc, P. Steckel, J. Menendez, G. Cisneros, and S. Jones, “Audience measurement modeling for convergent broadcasting and iptv networks,” Broadcasting, IEEE Transactions on, vol. 55, no. 2, 2009, pp. 502–515.

[3] F. Alvarez, D. Alliez, C. Martin, J. Menendez, and G. Cisneros, “Audience measurement technologies for user centric media,” in Consumer Electronics, 2008. ISCE 2008. IEEE International Symposium on, 2008, pp. 1–4.

[4] V. Gopalakrishnan, R. Jana, R. Knag, K. Ramakrishnan, D. Swayne, and V. Vaishampayan, “Characterizing interactive behavior in a large-scale operational iptv environment,” in INFOCOM, 2010 Proceedings IEEE, 2010, pp. 1–5.

[5] C. Costa, C. Ramos, I. Cunha, and J. Almeida, “Genius: a generator of interactive user media sessions,” in Workload Characterization, 2004. WWC-7. 2004 IEEE International Workshop on, 2004, pp. 29–36.

[6] T. Qiu, Z. Ge, S. Lee, J. Wang, J. Xu, and Q. Zhao, “Modeling user activities in a large iptv system,” in Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference, ser. IMC ’09. New York, NY, USA: ACM, 2009, pp. 430–441.

[7] S. d. C. A. Basilio, M. F. Moreno, and E. Barrére, “Supporting interaction and audience analysis in interactive tv systems,” in Proceedings of the 11th European Conference on Interactive TV and Video, ser. EuroITV ’13. New York, NY, USA: ACM, 2013, pp. 23–30.

[8] J. Norris, Markov Chains, ser. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1998.