Evaluation of Encoding and Network Aspects on Video Streaming Performance: A Modeling and Experimental Approach

Jamilson Dantas*, Rubens Matos†, Carlos Melo*, Jean Araujo‡, Joao Ferreira*, and Paulo Maciel*
* Informatics Center, Federal University of Pernambuco, Recife, Brazil
† Federal Institute of Education, Science, and Technology of Sergipe, Lagarto, Brazil
‡ Academic Unit of Garanhuns, Federal Rural University of Pernambuco, Garanhuns, Brazil
{jrd, casm3 jfsj3, prmm}@cin.ufpe.br*
rubens.junior@ifs.edu.br†, jean.teixeira@ufrpe.br‡

***Abstract*—The adoption of stochastic models has been one of the central topics in various architectures. One important step to adopt it is model validation, which aims at obtaining reasonable models to represent actual behavior of services components, it has been essential to validate models against actual measurements. System-wide model simulation results can be compared with recordings from the measurement. In this paper, we accomplish the model validation to Stochastic Petri Net (SPN) models created to evaluate VoD system hosted on a private cloud system, considering MP4, MPG, Ogg and FLV formats. We proposed the performance model to represent packet transfers, and to compute performance metrics, such as throughput, packet loss, and service delivery reliability. The SPN model enables a compact representation of a large number of packets generated by video streaming. We validate the models through experimental data, using a VoD streaming service in a cloud infrastructure testbed. We demonstrate that the proposed models are accurate and can be utilized for planning the quality of service (QoS) of corporate video streaming infrastructures. A case study is presented to compare the behavior of the system. Results indicate that the model validation way adopted can be a good solution for models validation. A case study is conducted to compare the behavior of the system under distinct network scenarios. The results indicate that video streaming QoS under 3G (EVDO) networks is significantly worse than other wireless technologies, such as WiFi, 3.5G (HSPA+), and 4G (LTE).**

***Index Terms*—Petri Nets, Model Validation, Quality of service**

I. Introduction

Most popular online services, such as social networks and video streaming servers, use cloud computing to ensure availability and proper service delivery to users [1]. Such applications usually require large data center infrastructures, a high availability of resources such as CPU, memory, storage, and network bandwidth. Current video streaming services offer a large variety of multimedia content in many formats, so the user can watch what he wants, with the proper resolution or quality according to his preferences and connection speed. Companies such as Netflix [2] and Hulu [3] have made consistent investments in this type of service. In fact, Netflix was pointed out as the single largest source of Internet traffic in the United States, consuming 29.7% of peak downstream traffic [4].

The video format adopted in the streaming service and the type of network connection might play an important role on the reliability and performance of VoD services. Models allow a comprehensive and thorough analysis regarding the effects caused by the parameters defined in the equations on the application. In addition, it can also establish possible relationships between each of the parameters. To validate the results achieved through the elaborate models, analytic technique can compare them to the actual values measured in experimental tests.

Some works present interesting studies about validation techniques. A model validation is performed in [5] with hybrid dynamic simulation to power engineering systems. In [6] is presented the development and implementation of an application and methodologies to validate the Energy Management System model in which model validation provides a mean to compare the archived State Estimation, Power Flow and Contingency Analysis results against the archived telemetry data. In another hand, some works present interesting results about Video streaming services. In [7], the authors study the performance of an adaptive video on demand (VoD) system. In that system, channel resources are allocated more efficiently than in full VoD in which a separate channel is allocated for each user or a fixed set of movies is played at regular time intervals, respectively. The authors evaluate this model VoD system using stochastic Petri nets and use discrete event simulation to determine the system performance, i.e., the number of requests accepted by the video server compared to the number of requests offered. In [8], the authors studied the packet loss in a video transfer over UDP with *ns-2* simulations driven by genuine video traffic.

In this paper we propose an SPN model to quantify the packet loss ratio, service reliability to end users and throughput, considering distinct video formats. Non-exponential distributions are allowed in this model using the moment matching technique. In the testbed environment, we use an open-source streaming application VLC hosted in a private cloud, providing videos in MP4, MPG, Ogg, and FLV formats (HEVC/H265, H264 - MPEG-4, Theora, and H264 - MPEG-4 codecs compression respectively). The use of the same codec (h264) in a different format it was made of purpose to compare the containers. Network traffic is captured and analyzed, providing input data for the SPN model and thus empowering results validation. A case study is adopted to demonstrate the service reliability on 3G, 3.5G, 4G, and WiFi networks. Specifically, our contributions are the following:

- a testbed methodology to evaluate a video streaming scenario using four video types, MP4, MPG, OGG, and FLV;
- a quantitative evaluation of the behavior of differents of videos type;
- development of SPN and CTMC models for the packet transfer of video stream;
- development of a closed-form equation to estimate the packet arrival probability for each video type in different network scenario.

The remaining of the paper is organized as follows. Section II shows preliminary concepts of video streaming service. Section III describes the system analyzed in this paper, the methodology and metrics adopted and the testbed setup. Section IV presents the proposed models. Section V presents the validation steps and numerical results conducted for distinct video formats transmitted through wireless network technologies and, Section VI draws some conclusions and points to possible future works.

II. Video Streaming Service

Streaming is a technique that enables multimedia (i.e., composition of two or more media) information consumption by a user while data is being transferred. Video streaming is a multimedia streaming where the video is continuously received and displayed to the end user. In short, the user can watch a video while later scenes are yet being transmitted. Thus he does not need to wait for the complete file transfer to begin watching. Video Streaming quality depends mainly on its encoding, protocol, and buffering mechanisms [9] [10], besides enough network bandwidth.

Many devices are used to consume video streaming nowadays, only requiring compatibility with the video format and network connectivity. Video streaming allows a multitude of content to be available to people around the world at a low cost and differentiated programming.

The following components are present in the architectural environment for video streaming: Video compression, application-layer QoS control, continuous media distribution services, streaming servers, media synchronization mechanisms, and protocols for streaming media [11]. In such environments, the audio and video data are pre-compressed using specific compression algorithm and then saved in storage devices [11]. Upon the clients request, the streaming server retrieves the compressed audio and video data from a storage device and after that packetize the compressed bit streams and sends the video/audio to the Internet to serve the user request [11].

III. Baseline Architecture and Validation Methodology

This section presents the cloud-enabled VOD system that was used for the testbed experiments in this work, as well as the baseline architecture that guided the construction of our stochastic models. The methodology applied in this study to create the models and validate them is also described further.

The Figure 1 depicts the video streaming testbed implemented in a private cloud. The cloud infrastructure is based on the Eucalyptus platform version 3.3.2 and the CentOS 6.4 GNU/Linux operating system. In this basic architecture, one node is available for deploying the virtual machines, and one volume supports video storage. Users request video stream via the web browser.

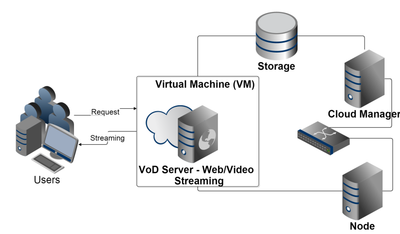


Fig. 1: VOD Baseline architecture

Wireshark network sniffer [12] is adopted to monitor outgoing VoD server traffic and incoming traffic in the user machine. We measured three types of data, which are useful for building the SPN model: the packet loss probability, mean time between video packets, and average network latency. The packet loss probability is the probability that a UDP packet is dropped along the path from sender to receivers, which might occur due to many reasons, such as interference, and congestion on switches and, routers. The mean time between video packets is a measure of the average time interval between consecutive network packets generated by the VoD server. The network latency is a measure of time that each packet takes on the network until it is received by the client.

The Figure 2 shows the activity diagram of the methodology applied in our study. There are five activities: problem definition, identification of interest metrics, acquisition of system information, model definition, and model validation. The first activity is the problem definition, and corresponds to the questions: What to evaluate? For what? And what are the expected results?

The second activity should provide the set of metrics to be evaluated. The most important metrics in our particular case are the packet loss, video packet throughput, amount of packets received by client and reliability, i.e., the probability of the service being entirely delivered.

The acquisition of system information (see Figure 3) consists of three steps. First, the Wireshark monitoring tool is started on both sides, client, and server. Next, the video stream

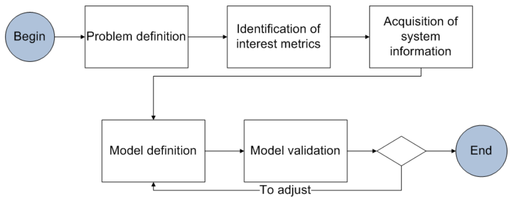


Fig. 2: Proposed methodology

is requested by a remote client. Next. It is important to stress that the clock of both computers (client and server) were synchronized with an NTP server from the Informatics Center of Federal University of Pernambuco (UFPE). The video streaming proceeds until all content have been transferred. When the video streaming ends, the network monitoring is stopped.

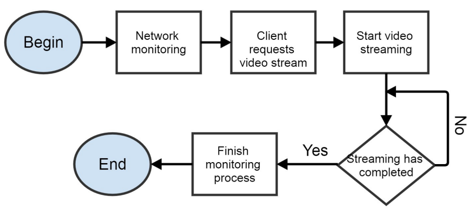


Fig. 3: Acquisition of system information

The fourth activity of the methodology depicted in Figure 2 is to create an SPN model, and this model is validated in the last activity, by comparing the model results with the data from experiments performed in the actual system. If the model results are not satisfactory, adjustments are made to represent the system characteristics in a better way.

For model validation purposes, the simulated model of cloud VoD system can be compared with the real system. It should be pointed out that both the simulated and real metrics are strictly associated with the same set of video formats and kind of data. Also, the inputs of the model and real systems are equally evaluated. If the model results are not satisfactory, adjustments are made to represent the system characteristics in a better way.

The model validation procedure is summarized as follows:

1) pre-define a set of input data to video format, size and virtual machines configuration
2) the experiments are executed in real system environment expending enough time to guarantee the statistical quality of data
3) the model previous conceived is analyzed and the metrics are computed
4) the provided results obtained from model and real system are compared
5) model validation results can then be further used for mode calibration

IV. Model definition

This section presents the proposed SPN model and the validation of its results through comparison with real system measures.

A. Time to Transfer Packets Model

We derived closed-form equations to evaluate the mean time for video transmission (see Equations 1, 2 and 3). For this analysis, we adopted an absorbing state Markov Chain model, depicted in Figure 4. The CTMC model represents the packet transmission, since it is assembled by the server (state Pkt_to_send) until it arrives at the client (state $Pkt\_arrived$). If some error occurs and the packet is not properly sent, the model goes to absorbing state Pkt_err. The probability of successful transmission is ps, whereas the outgoing packet throughput on server is λ_s. If there is no error, the model goes to state Pkt_sent. From Pkt_sent, the transition to the absorbing state $Pkt\_Arrived$ has the rate λ_L, denoting the reciprocal of network latency. All parameter have the same values indicated in Table I.

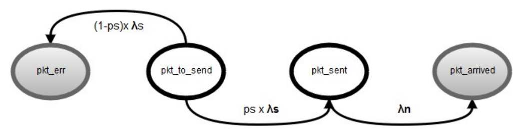


Fig. 4: CTMC model for packet transmission

$$P_{Pkt\_err}(t) = -te^{-\lambda_s}(-1+p_s)Sinh[\lambda_s], t > 0, \quad (1)$$

$$P_{Pkt\_sent}(t) = \frac{ps(\lambda_n - e^{-t\lambda_s}\lambda_n + (-1+e^{-t\lambda_n})\lambda_s)}{\lambda_n - \lambda_s}, t > 0. \quad (2)$$

Equation 1 computes the unsuccessful probability (Pkt_err) at time t, and the Equation 2 computes the successful probability (Pkt_sent) at time t. $Sinh[\lambda_s]$ represents the hyperbolic sine of λ_s. The mean time to transfer a packet (MTTP) is denoted by Equation 3. It is important to stress that the choice of CTMC model in this scenery occurs because there is the possibility to extract the closed-form equation.

$$MTTP = \frac{ps}{\lambda_L} + \frac{1}{\lambda_s}. \quad (3)$$

B. Performance Models

State-based models (e.g., stochastic Petri nets [13]) represent the system behavior (failure/repair activities, and performance issues). These models allow representing complex relationships between system components, such as dependencies involving sub-systems and resources constraints [14]. SPN supports simulation to solve models that might have a large, or even infinite, state-space.

Due to a big number of state that can be produced in a packet transfer network, we adopted an SPN model to evaluating the packet loss and system congestion. Figure 5 represents the conceived model. Notice that the model fits our needs, but it is not bound to any specific application domain.

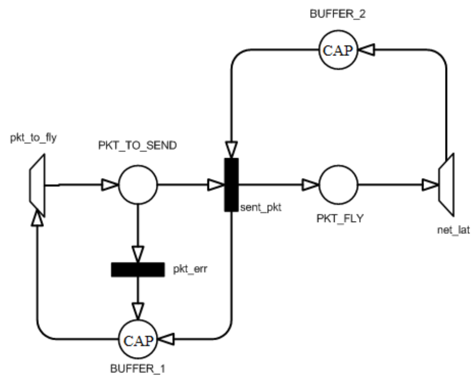


Fig. 5: Performance model

Therefore, it might also be used to analyze other network systems which are not related to the VoD service.

The initial transition ptk_to_fly is enabled, representing the packets transmission to the client when requested, generating the data traffic. The ptk_to_fly represents the inter-packet time and is assigned to an expolynomial distribution.

TABLE I: Definition of delays for timed transitions and weights for immediate transitions

	MP4	MPG	Ogg	FLV
W(pkt_err)	0.0029	0.0021	0.0027	0.0007
W(pkt_sent)	0.9971	0.9979	0.9973	0.9993
{pkt_to_fly}(ms)	5.748	6.813	5.8813	5.727
{net_lat}(ms)	28.236	26.951	25.981	25.683

The transition pkt_to_fly enabling depends on the network capacity, the place $BUFFER\_1$ represents that. This place represents the available bandwidth of the local area network (LAN), so the transition pkt_to_fly can fire only if there is one token available in such a place. The same behavior happens with the $sent\_pkt$ transition, that is enabled when there are tokens in places PKT_TO_SEND and $BUFFER\_2$, the latter representing the wide area network (WAN) available bandwidth. The transition pkt_err represents a packet loss by errors such as collisions and discards due to full buffer. The transition $sent\_pkt$ represents the proper transmission of the packet, considering that no errors occurred and there is enough bandwidth capacity available. When the place PKT_FLY receives one token, the timed transition net_lat is enabled, and it fires after the packet interarrival time elapses, This value was measured with the *Wireshark* tool, as mentioned for the methodology in Section III, and is presented in Table I. The values assigned to all transitions were measured in the real system testbed, as depicted in Figure 2. Such a table also shows the delay for the pkt_to_fly transition, and the weights assigned to the immediate transitions pkt_err and $sent\_pkt$. Those values were measured for the video formats MP4, MPG, Ogg, and FLV, with an average computed from 30 experiment runs.

Table II shows the expressions used to compute the target metrics with the SPN. Hereafter, SP, RP, and PL represents the amount of sent packets, received packets, and lost packets respectively. TC represent the throughput at the client side. T_v is the time duration of the video requested. The notation $E\{\#PKT\_FLY\}$ indicates the expected (i.e., average) number of tokens in the place PKT_FLY.

TABLE II: Expression for computing each metric on SPN

Metrics	Equation
SP	$TS \times T_v$
RP	$TC \times T_v$
PL	$SP - RP$
TC	$E\{\#PKT\_FLY\} \times (1/net\_lat)$

The trapezoid notation was adopted in the SPN model to represent expolinomial distributions for the transitions pkt_to_fly and net_lat. The moment-matching approach enabled us determining which expolynomial distribution best fits the packet interarrival and latency times, based on data collected through experimental testbeds. The respective mean (μ) and standard deviations (σ) were analyzed, and these transitions were refined. The Hyperexponential and Erlangian distributions are the expolynomial distributions most suitable to the measured packet interarrival time and networking latency respectively.

V. Model Validation and Numerical Results

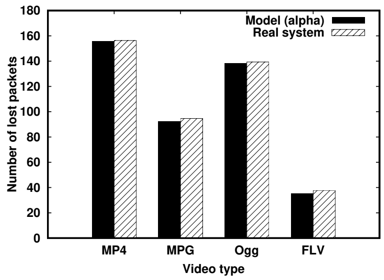
This section presents the model validation obtained through a rigorous comparison of model simulation results with observed behavior in a real system and makes it easier to identify problematic models and, numerical results obtained by a case study.

A. Performance Model Validation

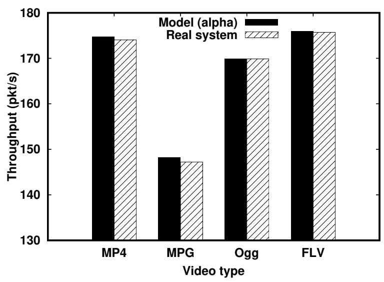
The validation is accomplished by comparing the results obtained from the SPN model, through analytical modeling, to measures collected directly from the testbed system. Four video formats were employed: MP4, MPG, OGG and FLV. Each video has a duration of five minutes, and the experiment was executed thirty times for each video. Figure 6(a) to 6(d) presents the comparison analysis for performance model validation.

Those Figures shows the total packet loss (Figure 6(a)); the effective throughput on server (in packets per second - (Figure 6(b))); the number of packets received by client (Figure 6(c)); and the ratio between the number of packets received and the total expected (Figure 6(d)) i.e., the percentage of the video stream packets that was properly received by the client. The number of packets received is not used here to compare directly the quality of service achieved with both video types, because each type uses distinct encoding techniques that yield different file sizes. Data from Figure 6(d) shall be employed for such a purpose, instead, since it presents the ratio between received and sent packets.

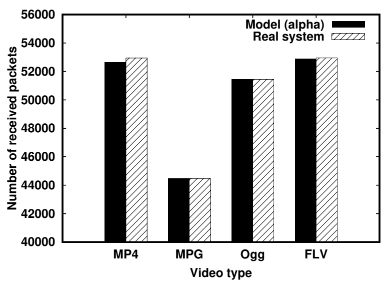
All four graphs show that model and real system results are close to each other, but we must confirm that with a suitable statistical method. To verify whether data from the model and real system are statistically equivalent, we adopt a nonparametric method called bootstrap [15]. This method



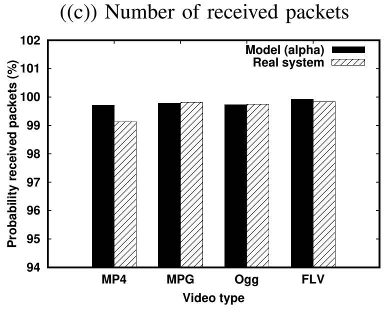
((a)) Packet Loss



((b)) Effective Throughput



((c)) Number of received packets



((d)) Percentage of Packets Received

Fig. 6: Comparison of model and real system results

enables the computation of confidence interval for all measures obtained in the experimental testbed. The results from analytical modeling fit within the confidence interval of the testbed experiments. Therefore, we can rely on model results as acceptable estimates of the real system behavior.

We may draw some important conclusions from both model and experiments results. Based on this analysis we can say that there is no statistical evidence to reject the hypothesis of equivalence between the real system measures obtained and the performance model.

B. Case study

This section presents a case study to evaluates the packets received by client, lost packets, and throughput in four wireless network technologies: WiFi (validation scenery), 3G (EVDO), 3.5G (HSPA+) and 4G (LTE). For this case study, the values of network latency and packet loss ratio for 3G, 3.5G, and 4G were obtained from [16], and were used to compute the performance metrics for the VoD streaming service in such network types. For WiFi scenario we adopted the values shown in Section IV Table I. Table III presents the parameters definition for 3G, 3.5G and 4G networks.

TABLE III: Parameters Definition for 3G, 3.5G and 4G Networks

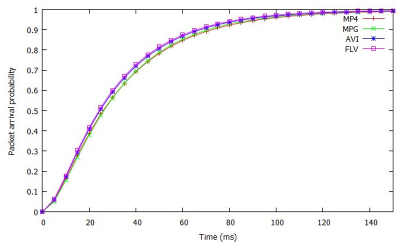
Technology	Latency (ms)	Loss Ratio (%)
4G LTE	77.31	0.13
3.5 HSPA+	115.32	0.14
3G EVDO	348.02	0.27

Note that the values presented in [16] are for 100MB file, not specifically for video files, therefore it is necessary to estimate the packet loss ratio for the four studied types of videos. We used the loss ratio measured in WiFi network for each type of video and the values obtained in Table III to derive the 3G, 3.5 and 4G-specific ratios. Seeing the biggest value of loss ratio of video in WiFi scenario obtained by MP4, we conducted the study considering that the MP4 type has the same loss ratio shown in [16] for each network type (3G, 3.5G, and 4G). For other types of video we used the proportion in relation to the MP4 type, equivalent to the proportion observed in the WiFi network, since MP4 was the type with greatest packet loss ratio. Thus for the MPG loss ratio on the 3G network, for example, we can calculate $P_{3G}^{MPG} = P_{3G}^{MP4} \times \alpha_{MPG}$. Table IV presents the loss ratio for each video type on each network type, and also the value of α used in the equation just presented.

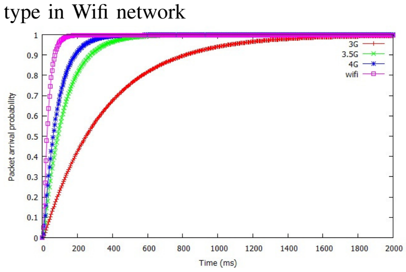
TABLE IV: Loss ratio for each video type on each kind of network connection

Video Type	P_{WiFi} (%)	α	P_{3G} (%)	$P_{3.5G}$ (%)	P_{4G} (%)
MP4	0.29	1	0.27	0.14	0.13
MPG	0.21	0.72	0.19	0.10	0.09
AVI	0.27	0.93	0.25	0.13	0.12
FLV	0.07	0.24	0.06	0.03	0.03

In Figure 7(a), we can see the probability of packet arrival at the destination for each video type in WiFi network. Note that the four video formats have a similar behavior, reaching nearly 100% probability at 140 ms, denoting that despite the differences in packet loss, the delay for reception with WiFi is almost the same for all video formats, not being impacted by processing, queuing, or packetizing differences on server side. Therefore, the choice of video formats in WiFi networks is not so relevant in terms of performance for video buffering and fast playback, for example. This choice could then depend on other aspects such as user preferences or video player software constraints.



((a)) Packet arrival probability for each video type in Wifi network



((b)) Probability for Each Scenario - used by MP4

Fig. 7: Comparison of packet arrival probability

Figure 7(b) depicts the transmission success probability for one packet at each scenario with a MP4 video type, which had the worst packet loss ratio. Figure 7(b) may also be interpreted as the distribution function of packet latency in the MP4 scenarios.

In Figure 7(b), it is noteworthy that the transmission success probability with WiFi reaches the highest levels sooner than with all other network types. This happens because WiFi has the smallest average latency among the studied technologies. Whereas WiFi reaches 100% probability in about 150 ms, 4G takes about 300 ms, and 3.5G about 500 ms. The worst performance occurs in 3G network, which reaches about 100% probability of packet arrival above of 1000 ms, and has a function much more dispersed throughout the middle and high latency values than the other network technologies. This shows that 3G is significantly less adequate for video streaming transmission than WiFi, 3.5G, and 4G.

VI. Final Remarks

A model validation technique is presented in this paper. Its validity has been illustrated by successful applications to VoD cloud systems and load model validations. Scenarios were adopted to compared the system values obtained through experimentation and by models analysis. To system-wide model validation, the proposed methodology has the following advantages:

- The proposed methodology enables rigorous direct comparison of simulation and measurements because the same stimulus (metrics analysis) is applied for the simulation and in the measurement.
- It is easier to interpret the model validation results and identify problematic models within a much smaller simulation subsystem.

We proposed a methodology based on testbed monitoring and analytical modeling to validate models and evaluate some metrics of a video streaming service. We used a private cloud environment to build the testbed and perform the experiments. We also proposed an SPN model which requires as input the packet loss probability, maximum network capacity, and the average network latency. The results, obtained through stationary analysis, are statistically equivalent to the results obtained by measurements from the system under test. We also employed a CTMC model for deriving closed-form expressions to calculate some metrics for this system.

In future, we intend to extend our study considering the generalized closed-form equation to compute the packet arrival probability independent of file kind. However, we will have to expand the testbed methodology. We are also intent on extending the models for analyzing the performability of the system considering the VM type in cloud infrastructure.

References

[1] W. Zhu, C. Luo, J. Wang, and S. Li, "Multimedia cloud computing," *Signal Processing Magazine, IEEE*, vol. 28, no. 3, pp. 59–69, 2011.

[2] Netflix. (2017) Netflix - watch tv shows movies anytime, anywhere. Netflix.com, Inc. Available in: https://www.netflix.com/.

[3] Hulu. (2017) Hulu - watch tv, watch movies. Available in: https://www.hulu.com/.

[4] V. K. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z.-L. Zhang, "Unreeling netflix: Understanding and improving multi-cdn movie delivery," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 1620–1628.

[5] Z. Huang, D. Kosterev, R. Guttromson, and T. Nguyen, "Model validation with hybrid dynamic simulation," in *Power Engineering Society General Meeting, 2006. IEEE*. IEEE, 2006, pp. 9–pp.

[6] F. Ahmad and M. A. Nayeem, "Real time power system model validation using archived data," in *Southeastcon, 2012 Proceedings of IEEE*. IEEE, 2012, pp. 1–6.

[7] O.-M. Penttinen, T. Pyssysalo, and L. Ojala, "Performance analysis of adaptive video on demand using stochastic petri nets," in *System Sciences, 1997, Proceedings of the Thirtieth Hawaii International Conference on*, vol. 5. IEEE, 1997, pp. 119–124.

[8] V. Markovski, F. Xue, and L. Trajković, "Simulation and analysis of packet loss in user datagram protocol transfers," *The Journal of Supercomputing*, vol. 20, no. 2, pp. 175–196, 2001.

[9] A. Thilagavathy, K. Aarthi, and A. Chilambuchelvan, "A hybrid approach to extract scene text from videos," in *Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on*. IEEE, 2012, pp. 1017–1022.

[10] V. T. Chasanis, C. Likas, and N. P. Galatsanos, "Scene detection in videos using shot clustering and sequence alignment," *Multimedia, IEEE Transactions on*, vol. 11, no. 1, pp. 89–100, 2009.

[11] D. Wu, Y. T. Hou, W. Zhu, Y.-Q. Zhang, and J. M. Peha, "Streaming video over the internet: approaches and directions," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 3, pp. 282–300, 2001.

[12] Wireshark. (2017) Wireshark. Available in: http://www.wireshark.org/.

[13] M. K. Molloy, "Performance analysis using stochastic petri nets," *Computers, IEEE Transactions on*, vol. 100, no. 9, pp. 913–917, 1982.

[14] P. Maciel, K. S. Trivedi, R. Matias, and D. S. Kim, "Dependability modeling," in *Performance and Dependability in Service Computing: Concepts, Techniques and Research Directions*. Hershey: IGI Global, 2011.

[15] R. Pereira, M. Azambuja, K. Breitman, and M. Endler, "An architecture for distributed high performance video processing in the cloud," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*. IEEE, 2010, pp. 482–489.

[16] Y.-C. Chen, D. Towsley, E. M. Nahum, R. J. Gibbens, and Y.-s. Lim, "Characterizing 4g and 3g networks: Supporting mobility with multipath tcp," *School of Computer Science, University of Massachusetts Amherst, Tech. Rep*, vol. 22, 2012.