

Gaussian semi-Markov Model based on Real Video Multimedia Traffic

Scott Fowler, Jalal Sarfraz, Muhammad Muddassir Abbas and Vangelis Angelakis

Linköping University Post Print



N.B.: When citing this work, cite the original article.

©2015 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Scott Fowler, Jalal Sarfraz, Muhammad Muddassir Abbas and Vangelis Angelakis, Gaussian semi-Markov Model based on Real Video Multimedia Traffic, 2015, IEEE International Conference on Communications Communications (ICC) 2015.

Postprint available at: Linköping University Electronic Press

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-117277>

Gaussian semi-Markov Model based on Real Video Multimedia Traffic

Scott Fowler, Jalal Sarfraz, Muhammad Muddassir Abbas and Vangelis Angelakis
Mobile Telecommunications, Department of Science and Technology, Linköping University, Norrköping, Sweden

Abstract—The 3rd Generation Partnership Project (3GPP) introduced the new radio access technology, LTE (Long Term Evolution) and LTE-Advanced, which has the capability to provide larger bandwidth and low latencies on a wireless network in order to fulfill the demand of Users' Equipment (UEs) with acceptable Quality of Service (QoS). One of the data-heavy applications that has exploded on the market is Video. This calls for accurate modeling of video traffic. To ensure the quality and correctness of complex systems with Video traffic, it is important to evaluate the behavior of the traffic in the heterogeneous environment. This paper presents measurements on LTE Video traffic which were performed in a real environment. We derived a semi-Markov model which accurately reproduces the statistics of composite Video measurements over LTE network.

I. INTRODUCTION

The mobile-only data traffic is expected to increase 11-fold by 2018 [1]. While emails, text-messages, and Voice-over-IP (VoIP) calls will remain to be important communication means for mobile data, we are at the threshold to embrace the next big change; Video Conferencing. Video traffic is expected to be more than half of the internet traffic by 2018, resulting in greater bandwidth usage on the limited wireless channels, since Video calls are much more bandwidth-demanding than VoIP calls.

As a result, there has been an increasing interest in streaming Video over wireless networks, and the trend is expected to continue. This phenomenon is mainly attributed to the advancements of fourth generation (4G) network, the resulting wide availability of high-speed internet access, the growing market of mobile entertainment, and the increasing number of web sites which offer their users high-quality multimedia content. Even though there is already a significant global roll out of LTE, little is known about the usage behavior and the traffic characteristics of the LTE mobile devices in heterogeneous environment. Unlike files transfers, a Video can tolerate some packet/frame losses, but it is delay sensitive. Therefore it is important to accurately analyze Video over wireless networks. It is of great interest for end users and application developers to assess the quality of Video calls under different network conditions. An effective method to achieve this content analysis.

Content analysis of Video is to find meaningful structures and patterns from raw data by putting them in a compact form such that efficient indexing, classification, and retrieval of Video behaviour. Early works mainly focused on Video shot boundary detections [2]. For indexing and retrieval of Video shots for boundary detections, key-frames will be selected to

represent the Video. Besides doing Video temporal segmentation, localized features such as object-based representation of Video have also been investigated by researchers. In [3], the trajectories of objects based on object segmentation and tracking were used for Video indexing. However, these analysis only provides a very small time scale access and low level analysis for Video. Also, it is assumed that the content of a Video can be analyzed.

As major manufacturers migrate, they have been poised to provide Video services. The fact that many distributed Video calls are offered free-of-charge, further fueled the wide adoption of Video conferencing. Up to now, there have been very few study on the Video telephony applications [4]. Today, Skype [4], [5], is the most popular Video communication platform for two-way Video calls and multi-party Video conferencing. It has become urgent need to model Video behavior, due to this rapid growth trend of use of Video calls. This calls for accurate modeling of the statistical properties of Video calls with emphasis on efficient bandwidth and resource allocation to provide a dependable and reliable network architecture. In this paper, a real time cumulant Probability Density Function (*PDF*) using Video call and Live Video streaming measurements is utilized. A parametric method for cumulants estimation using the Gaussian Mixture Model (GMM) is derived. The cumulants are expressed in terms of the GMM parameters, and estimated using the Expectation Maximisation algorithm. We derive a semi-Markov model for Uplink and Downlink which accurately reproduces the statistics of composite real empirical Video measurements. The Video models is based on Skype Video and Live Video streaming empirical measurements over LTE equipment to formulate our proposed model.

II. OVERVIEW OF GAUSSIAN MIXTURE MODEL

Density estimation with Gaussian Mixture Models (GMM) is a popular generative technique, mostly because it is computationally tractable and often produces good results. A GMM is a Probability Density Function (*PDF*) represented as a weighted sum of Gaussian component densities. The mixture of Gaussians for background modeling was first proposed in the year 2000 [6]. However, the parameters in GMM cannot be estimated because of the training data involving hidden parameters. The parameters of this GMM were estimated by way of the Expectation Maximisation (EM) algorithm [7], [8], a procedure commonly used to estimate mathematical model parameters given that there is limited observed or incomplete

data to complete the model. One of the powerful properties of GMM is that it can smoothly approximate an arbitrary probability distribution with the linear combination of finite Gaussian distribution. Each component density is a D -variate Gaussian function of the form:

$$P(t_k|\theta_k) = \sum_{l_k=1}^{n_k} \phi_{k,l_k} \mathcal{N}(t_k|\mu_{k,l_k}, \sigma_{k,l_k}), \quad k = 1, \dots, K \quad (1)$$

where $t=(t_1, t_2, \dots, t_K)$ is a set of K points in D -dimensional continuous-valued data vector (in our case *time*), $\mathcal{N}(\cdot)$ denotes a Gaussian density function and n_k denotes the number of Gaussians. The mixing proportions are denoted by $\{\phi_{k,l_k}\}_{l_k=1}^{n_k}$, such that $\sum_{l_k=1}^{n_k} \phi_{k,l_k} = 1$. The means and variances of the Gaussians are denoted by $\{\mu_{k,l_k}\}_{l_k=1}^{n_k}$ $\{\sigma_{k,l_k}\}_{l_k=1}^{n_k}$. The vector distribution parameters of the k^{th} source is denoted by $\theta_k = \{\sigma_{k,l_k}, \mu_{k,l_k}\}_{l_k=1}^{n_k}$. By applying the assumption of independent, their joint *PDF* can be formulated as follows:

$$\begin{aligned} P(t|\theta) &= \prod_{k=1}^K P(t_k|\theta_k) \\ &= \sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \dots \sum_{l_K=1}^{n_K} \phi_{1,l_1} \phi_{2,l_2} \dots \phi_{K,l_K} \\ &\quad \cdot \mathcal{N}(t_1|\mu_{1,l_1}, \sigma_{1,l_1}) \mathcal{N}(t_2|\mu_{2,l_2}, \sigma_{2,l_2}) \\ &\quad \dots \mathcal{N}(t_K|\mu_{K,l_K}, \sigma_{K,l_K}) \\ &= \sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \dots \sum_{l_K=1}^{n_K} \omega_{l_1, l_2, \dots, l_K} \\ &\quad \cdot \mathcal{N}\left([t_1, \dots, t_K]^T \mid [\mu_{1,l_1}, \dots, \mu_{K,l_K}]^T, \right. \\ &\quad \left. \text{diag}(\sigma_{1,l_1}, \dots, \sigma_{K,l_K})\right) \\ &= \sum_{i=1}^M \omega_i \mathcal{N}(t|\mu_i, \sigma_i) \end{aligned} \quad (2)$$

$M = \prod_{k=1}^K n_k$ is the total number of Gaussians in the joint *PDF* and $\omega_i = \prod_{k=1}^K \phi_{k,l_k}$; $i = 1, \dots, M$ are the mixing proportions of each Gaussian such that:

$$\begin{aligned} \int_{\mathcal{R}^D} P(t|\theta) dt &= \int_{\mathcal{R}^D} \sum_{i=1}^M \omega_i \mathcal{N}(t|\mu_i, \sigma_i) dt \\ &= \sum_{i=1}^M \omega_i \int_{\mathcal{R}^D} \sum_{i=1}^M \mathcal{N}(t|\mu_i, \sigma_i) dt \\ &= \sum_{i=1}^M \omega_i = 1 \end{aligned} \quad (3)$$

where $t = (t_1, t_2, \dots, t_m) \in \mathcal{R}^D$ is a D -dimensional continuous-valued data vector (in our case Inter-arrival time (*IAT*)). The index i denotes a unique combination of Gaussians for *IAT*, i.e., $l_1, \dots, l_K \rightarrow i$, where $l_k \in [1, \dots, n_k]$

denotes a Gaussian index of the k^{th} *IAT*. The mean vector and standard deviation of the i^{th} Gaussian are denoted by $\mu_i = [\mu_{1,l_1}, \dots, \mu_{K,l_K}]^T$ and $\sigma_i = \text{diag}(\sigma_{1,l_1}, \dots, \sigma_{K,l_K})$, respectively. The complete GMM is parametrized by the mean vectors, standard deviation matrices and mixture weights from all component densities. These parameters are collectively represented by the notation:

$$\theta = \{\omega_i, \mu_i, \sigma_i\}_{i=1}^M \quad i = 1, \dots, M \quad (4)$$

We have an M -tuple of random *IAT* vectors $T=(T_1, T_2, \dots, T_M)$ that was generated in an independent and identically distributed (i.i.d) manner. We also assumed that the distribution of T depends on a parameters $\theta = (\theta_1, \dots, \theta_k) \in \Theta$. Therefore, for any $T_i \in T$, the individual *PDF* for $T_i \stackrel{iid}{\sim} P(t_i|\theta)$ for $i = 1, \dots, M$. By taking advantage of our sample being *i.i.d* in Equation (2) we will be able to say $t = (t_1, \dots, t_K)$ which denotes the observed sequence of outcomes of the random variables T_1, \dots, T_K . The *PDF* of the complete data $P(t|\theta)$, where $\theta \in \Theta$. Then given the observed values $T = t$, the expectation function is defined as:

$$\begin{aligned} \mathcal{L}(\theta|t) &= \mathcal{L}(\theta_1, \dots, \theta_n | t_1, \dots, t_n) \\ &= \prod_{k=1}^K P(t_k|\theta_k) \\ &= \sum_{i=1}^M \omega_i \mathcal{N}(t|\mu_i, \sigma_i) \end{aligned} \quad (5)$$

Note, when referring to $P(t|\theta)$, it is assumed that θ is some unknown fixed quantity, whereas t is allowed to vary over all possible values in our sample space. When the expectation function $\mathcal{L}(\theta|t)$ is in consideration we treat t as known, whereas now θ is allowed to vary over the parameter space Θ . The only distinction between the two functions is which variable is considered fixed and which is allowed to vary.

For an observed sample t let:

$$\hat{\theta}(t) \in \arg \max_{\theta \in \Theta} \mathcal{L}(\theta|t) \quad (6)$$

Here $\arg \max \mathcal{L}(\theta|t)$ denotes the set of all values $\theta \in \Theta$ which maximize $\mathcal{L}(\theta|t)$ over our parameter space Θ . Each component density is a D -variate Gaussian function form:

$$\mathcal{N}(t|\mu_i, \sigma_i) = \frac{1}{t \sigma_i \sqrt{2\pi}} e^{-\frac{(\log(t) - \mu_i)^2}{2\sigma_i^2}} \quad (7)$$

where $\mathcal{N}(t|\mu_i, \sigma_i)$ is the probability density in the mixture of with mean vector μ_i and σ_i standard deviation of Gaussian. The mixture weights satisfy the constraint that $\sum_{i=1}^M \omega_i = 1$. The complete GMM is parameterized by the mean vectors, standard deviation matrices and mixture weights from all component densities. To formulate, the density function of random variable t with a mixture of m Gaussian components expression is defined as:

$$P(t|\theta) = \sum_{i=1}^M \omega_i \mathcal{N}(t|\mu_i, \sigma_i) \quad (8)$$

where t is a D -dimensional continuous-valued data vector (in our case *time*), ω_i , $i = 1, \dots, M$, are the mixture weights, and $\mathcal{N}(t|\mu_i, \sigma_i)$, $i = 1, \dots, M$, are the component Gaussian densities.

Figure 1 illustrates the concept of merging mixture of Gaussian distribution into a Gaussian mixture distribution. Consider *State* j with an M -mixture Gaussian density. Since the mixture gain coefficients $\sum_{i=1}^M \omega_i = 1$, they are defined as a set of transition coefficient to substates *State* j_1 (with a transition probability ω_1), *State* j_2 (with a transition probability ω_2) through *State* j_M (with a transition probability ω_M). In each *State* j_M , there is a single mixture with μ_i and σ_i .

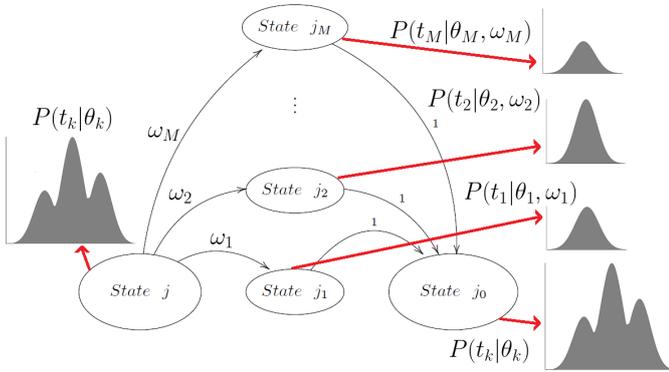


Figure 1: Overall structure of a GMM

From this GMM, we formulate a set of models for Uplink and Downlink. The EM algorithm is an iterative method for obtaining the parameters and re-estimating the parameter set. Another approach of inference strategy is the Maximum Likelihood Estimation (MLE). MLE gives good results in terms of efficiency but is too difficult to calculate. The model was developed based on the empirical measurement study of Skype Video calls and real-time Video (live web) streaming.

III. GMM PROBABILITY DENSITY MODEL

The observation density in each mixture depends on the substate (e.g. *State* $j_1, \dots, \text{State } j_M$). The method for combining the information of the substate variables is the observed distribution according to Gaussian *PDF* with a common variance and the mean which is a linear combination of the substate means. For $t \in \mathcal{R}^D$, we define D -dimensional multivariate continuous data vector by the parameter vector θ of mean μ and standard deviation σ . For the Video, call the log-normal model for the density is represented for the Uplink:

$$\mathcal{N}(t|\mu, \sigma) = \sum_{i=1}^3 \alpha_i \frac{1}{t\sigma_i\sqrt{2\pi}} e^{-\frac{(\log_{10}(t)-\mu_i)^2}{2\sigma_i^2}} \quad (9)$$

and for the Downlink:

$$\mathcal{N}(t|\lambda, \zeta) = \sum_{i=1}^3 \beta_i \frac{1}{t\zeta_i\sqrt{2\pi}} e^{-\frac{(\log_{10}(t)-\lambda_i)^2}{2\zeta_i^2}} \quad (10)$$

where α_i is the normalising weight, μ_i is the mean and σ_i is the standard deviation of each Gaussian component representing the density of the Uplink periods. For Downlink, β_i is the normalizing weight, λ_i is the mean and ζ_i is the standard deviation of each Gaussian component.

For the Live Streaming log-normal mixture model, the probability density equation is given where γ_i is the normalising weight, τ_i is the mean and ϕ_i is the standard deviation of each Gaussian component representing the density is given for the Uplink:

$$\mathcal{N}(t|\tau, \phi) = \sum_{i=1}^9 \gamma_i \frac{1}{t\phi_i\sqrt{2\pi}} e^{-\frac{(\log_{10}(t)-\tau_i)^2}{2\phi_i^2}} \quad (11)$$

and Downlink is:

$$\mathcal{N}(t|\psi, \varphi) = \sum_{i=1}^6 \delta_i \frac{1}{t\varphi_i\sqrt{2\pi}} e^{-\frac{(\log_{10}(t)-\psi_i)^2}{2\varphi_i^2}} \quad (12)$$

here, δ_i is the normalising weight, ψ_i is the mean and φ_i is the standard deviation of each Gaussian component representing the density of the Downlink periods.

A. Traffic Analysis Methodologies

All the raw data or network traces were processed in such a manner that we would be able to get maximum information out of it. Source and the destination IPs were identified along with their port numbers so as to visualize the flow of traffic between them. This data was further re-arranged in a manner to calculate the Inter-arrival Time (*IATs*). The *IAT* is the time between the successive packets arrived [9] at the device where calculations and measurements are intended to be performed. Typically, in traffic modeling and analysis, the packet sizes and the *IAT* between the packets are the two commonly used parameters. Instead in this paper, all the statistical calculations revolve around the *IATs* of the measured traffic streams.

It is important to mention that in some instances, in our generated *PDF* graphs, the density goes above 1. The reason is that we are not plotting the absolute or the relative frequency probabilities, but instead, probability densities; the ratio of the relative frequencies to the bin sizes. Since the data was so large, the bin sizes in some instances were so small that the resulting densities went beyond 1. But when we check the *CDF*, we see that the total probabilities add up to 1 which confirms that our results are correct [10]. Initially, Matlab codes that we ran produced *PDF* graphs that were discrete time in nature. But to get the general trend of the data, non-parametric density estimation was applied to convert them into continuous time. The non-parametric density estimation is explained in the following section.

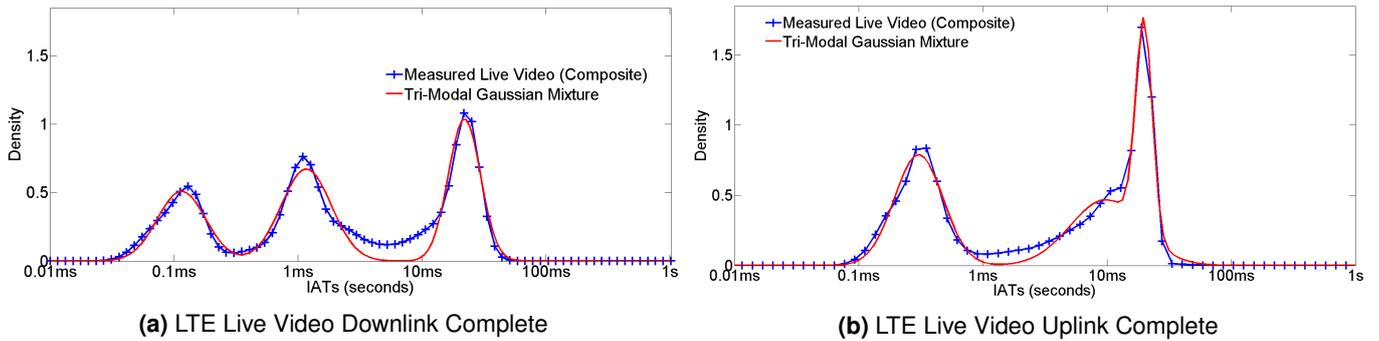


Figure 2: LTE Video Call with Raw Data and Fitted Data

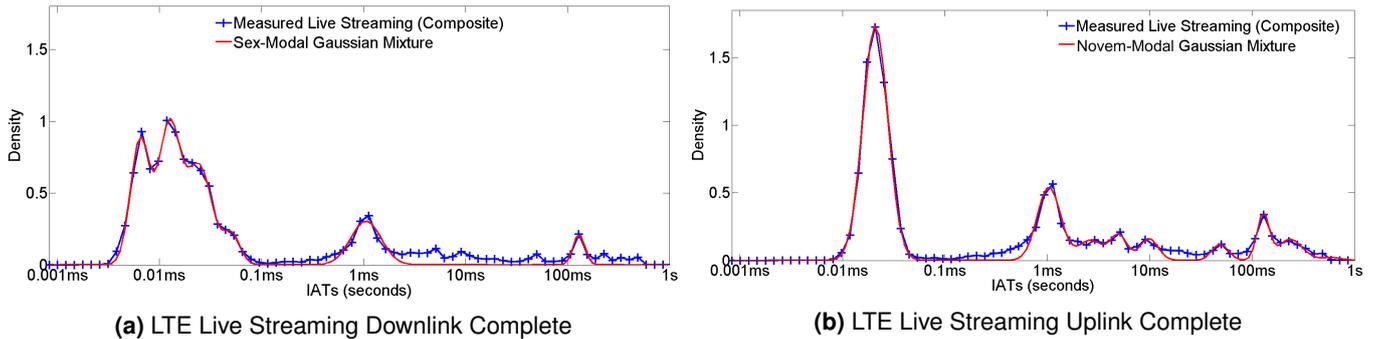


Figure 3: LTE Live Streaming with Raw Data and Fitted Data

B. Non-parametric density estimation

Initially, the processing of the raw data using Matlab produced distribution function graphs that were discrete time in nature. But to get the general trend of the data, non-parametric density estimation was applied to convert them into continuous time. Typically, non-parametric methods do not assume that the data has been taken from any particular probability distribution as opposed to the parametric methods which does assume that the data follows a particular probability distribution and consequently the parameters of the distribution. Instead, non-parametric, the techniques assume that the structure of the model is not fixed. Due to the lesser assumptions and the simplicity the methods are widely applicable than parametric methods. The detailed study of the non-parametric methods can be found in [11]–[13].

IV. TESTING ENVIRONMENT

The data collection was performed with the Samsung GT-B3730 USB dongle modem. The modem is capable of supporting LTE (2600MHz), 3G HSPA/UMTS (2100MHz), and 2G/EDGE (900/1800MHz) with the peak Uplink/Downlink speeds of 50/100 Mbps for LTE, 5.76/17 Mbps for HSPA/UMTS, and 107/296 Kbps for 2G/EDGE. The device came with its own drivers for Windows, Linux, and Mac operating systems. We collected the measurements for LTE by connecting the device to a desktop computer with an Intel Core 2 Duo 1.6GHz processor and 1GB RAM. The calling party was using an Intel Core i3 based laptop with 2GB RAM. The

Wireshark Network Protocol Analyzer v. 1.6.7 was used to capture network traces for Skype Video call and Live Video Streaming (LVS). Skype, v. 5.10.0.114 was installed on the desktop data collector which is the same Skype version on the calling party. A live football match was used for the LVS. Over a period of 3 months, a large amount of measurements were taken. Approximately 1 million packets per traffic type, allowed a sufficient amount of data to produce meaningful statistical results and conclusions.

V. RESULTS FROM THE TEST

Figures 2(a) and 2(b) depict the behaviour of LTE Live Video call for the Uplink or Downlink, respectively, where cross stitch lines represent distribution of the measured values, whereas smooth lines are the corresponding *PDF*. There are two significant peaks in Uplink and three in the Downlink Video stream. The significant *IAT* are: 1) 0.32ms and 18.96ms (there is a less significant at 11.8ms which will appear when calculating model) 2) 0.11ms, 0.95m and 18.96ms. When *IATs*

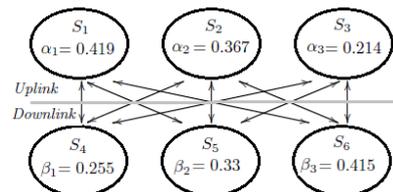


Figure 4: LTE Video call semi-Markov source model

Link	Parameters	Values		
		$i=1$	$i=2$	$i=3$
Uplink (S_1, S_2 and S_3)	α_i	0.419	0.367	0.214
	μ_i	-3.5173	-1.7023	-2.0115
	σ_i	0.1957	0.0657	0.2858
Downlink (S_4, S_5 and S_6)	β_i	0.255	0.330	0.415
	λ_i	-4.0004	-2.9942	-1.7187
	ζ_i	0.1941	0.2021	0.1295

Table I: LTE Skype Video Uplink and Downlink

for Uplink and Downlink data traffic are examined, 50% of the Uplink $IATs$ are equal or below 5.6ms whereas for Downlink, 50% of the $IATs$ are equal or less than 1.2ms. This is indicating that the Downlink data is the major contributor of the low $IATs$ of a combined traffic.

Figure 3(a) shows that the PDF for LTE live streaming for Downlink has significant peaks at 0.007ms, 0.009ms, 0.012ms, 0.017ms, 0.025ms, 0.045ms, 50.23ms and 129.62ms. Figure 3(b) for Uplink has noticeable peaks at 0.021ms, 1.12ms, 5.15ms, 9.10ms, 50.23ms and 129.63ms. The increased concentration of $IATs$ can be observed around 0.02ms, 1ms and 100ms for Uplink, whereas Downlink traffic has high concentration even below 0.1ms. Downlink clusters exist around 1ms and 100ms but the concentration is much lower than those of Uplink traffic. The observation that $IATs$ for the Downlink traffic in live streaming are generally low depicts more data is sent from the network to the user on the Downlink.

VI. SOURCE MODELS

Based on the fitted log-normal mixture distributions to the Uplink and Downlink periods from Equations (9) and (10) a simple 6-state semi-Markov model for the Video call was constructed (Figure 4). The three modes of the modal Log-normal distribution of the Uplink periods are represented by states 1, 2 and 3, respectively, whereas the three modes of the modal log-normal distribution of the Downlink periods are represented by the states 4, 5 and 6, respectively. The estimated values of these parameters are shown in Table I. For the Sex-Modal Gaussian for both the Uplink and Downlink, the performance evaluation of the proposed model-based technique was examined in terms of root-mean-square error (RMSE) and compared with the difference between the best-fit model and the raw real empirical data over the fitted segment. Uplink and Downlink, RMSE, were 2% and 5.02%, respectively.

This means the models are the Gaussian distributions with an efficiency/accuracy of 98% and 95%.

The same procedure for the Live Streaming was performed. We observed a bi-quint-modal profile for the density of Uplink and Downlink periods in the logarithmic domain as shown in Figure 5. The nine modes of the modal Log-normal distribution of the Uplink periods are represented by states 1-9, respectively, whereas the six modes of the modal log-normal distribution of the Downlink periods are represented by the states 10-15, respectively. The estimated values of these parameters are shown in Table II. According to the fitted log-normal mixture distributions to the Uplink and Downlink periods from Equations (11) and (12), the Uplink and Downlink RMSE were 1.64% and 3.31%, respectively. This means the models the Gaussian distributions with an efficiency/accuracy of 98.36% and 96.69%, respectively.

It should be noted that, we assume that the state transition probabilities are independent of the initial state which allows a unique assignment: the probability of visiting a state i is given by the normalising weights $\alpha_i, \beta_i, \gamma_i$ and δ_i in Equations (9) - (12), respectively, as shown in brackets in Figures 4 and 5.

VII. OVERALL STATISTICS OF THE VIDEOS

The overall statistics for the Video call and the Live Streaming are shown in Table III. From the Table, the Video Call the $\langle packets/sec \rangle$ are various similar in the number of packets. However, when the $\langle KBytes/s \rangle$ to average packet size $\langle Packet Size \rangle$ are examined, the Uplink is much smaller compared to the Downlink. This implies that there is considerably more overhead in the Downlink than in the Uplink connection of the video traffic. The Uplink and Downlink statistics shows that the mean for the Downlink is 7.2ms, while that for Uplink is 8.6ms. This implies the Downlink has higher data rate compared to the Uplink. When the standard deviations were compared, the variations for the mean of the Downlink $IATs$ was higher than that of the Uplink. These statistical results are in agreement with increased jitter experienced during our data collection for the Video call.

For the Live Streaming, the $\langle packets/sec \rangle$ on the Downlink was much greater than that of Uplink. Since in Live Streaming, most of the data is downloaded onto the user machine. This is evidenced by the $\langle Packet Size \rangle$ on Downlink being about ~ 17.5 times more. Hence, the data rate will be greater in the Downlink compare to the Uplink. The mean of the

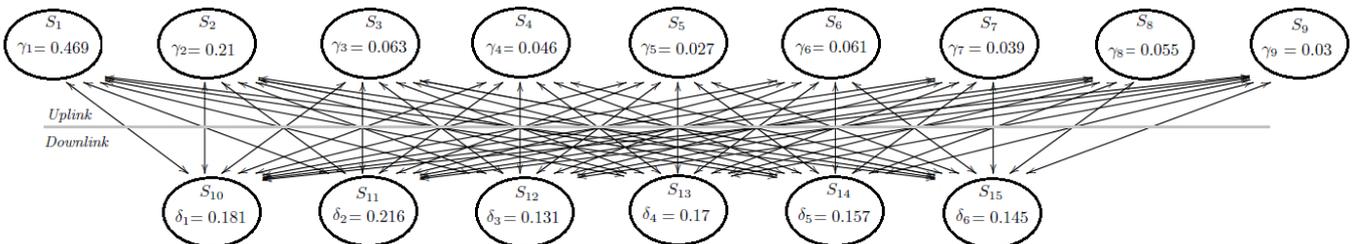


Figure 5: LTE Live Streaming Video semi-Markov model

Link	Parameters	Values								
		$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	$i=6$	$i=7$	$i=8$	$i=9$
Uplink (S_1, S_2, \dots, S_9)	γ_i	0.469	0.210	0.063	0.046	0.027	0.061	0.039	0.055	0.030
	τ_i	-4.6795	-2.9849	-2.5774	-2.3034	-2.0073	-1.3118	-0.8933	-0.6277	-0.2344
	ϕ_i	0.1222	0.1130	0.1178	0.0799	0.0881	0.0804	0.0590	0.1176	0.1097
Downlink ($S_{10}, S_{11}, \dots, S_{15}$)	δ_i	0.181	0.216	0.131	0.170	0.157	0.145	-	-	-
	ψ_i	-5.1900	-4.9028	-4.6108	-4.2928	-2.9763	-0.8876	-	-	-
	φ_i	0.0767	0.0918	0.0557	0.0722	0.0668	0.0617	-	-	-

Table II: LTE Stream Video Uplink and Downlink

Table III: Statistics of Video Call & Live Streaming

Statistics for the Video Call				
Parameters	$\langle IATs \rangle$ (ms)		σ of $\langle IATs \rangle$ (ms)	
Uplink	8.6		9.2	
Downlink	7.2		9.5	
Parameters	$\langle Packets/s \rangle$	$\langle K Bytes/s \rangle$	$\langle Mbps/s \rangle$	$\langle Packet Size \rangle$ (Bytes)
Uplink	116.4	33.0	0.264	283.9
Downlink	139.4	118.1	0.945	847.4
Statistics for the Live Streaming				
Parameters	$\langle IATs \rangle$ (ms)		σ of $\langle IATs \rangle$ (ms)	
Uplink	24.4		83.2	
Downlink	13.1		60.0	
Parameters	$\langle Packets/s \rangle$	$\langle K Bytes/s \rangle$	$\langle Mbps/s \rangle$	$\langle Packet Size \rangle$ (Bytes)
Uplink	40.98	2.612	0.021	63.76
Downlink	76.05	86.6	0.693	1138.9

$IATs$ for the Downlink is lower than the mean of the $IATs$ for Uplink, which is consistent with the fact that continuous stream of packets with minimum packet delay needs to be sent on the Downlink in Live streaming. Moreover, the σ of the $IATs$ on the Downlink is reasonable when compared to the corresponding mean, which implies Live streaming experiences more latency and jitter compared to Video call.

VIII. CONCLUSION

We have presented accurate measurements of the Uplink and Downlink of LTE for Video multimedia. Our study on the impact of Video multimedia revealed that the live Video streaming had more peaks compared to the Video call. The increased number of the peaks informs us that live Video streaming is a more bursty traffic compared to the Video call. Further, the analysis of the composite samples revealed the heavy tailed and multi-modal log-normal nature of the Uplink and Downlink periods. When we look into the Video call for Uplink and Downlink parts, for the Uplink, 50% of the $IATs$ are equal or below 5.6ms, while in the Downlink, 50% of the $IATs$ are equal or less than 1.2ms, which indicates that the Downlink data is contributing to the low $IATs$ of a combined traffic. In the Live Video Streaming, the increased concentration of $IATs$ is observed to be around 0.02ms, 1ms and 100ms for Uplink, while Downlink traffic has high concentration below 0.1ms and there are clusters around 1ms and 100ms but the concentration is very low as compared to the Uplink traffic. Thus, the $IATs$ for the Downlink traffic in live streaming are generally low as most of the data is sent from the network to the user on the Downlink. Since mathematical modeling of the data traffic is effective in analysis of the specific parts of the data in a distributed manner, the knowledge obtained through the process is especially useful for identifying bottlenecks in LTE

networks. This is particularly crucial for wireless networks where spectral economy is vital. Conventional Markov source models with exponential Uplink and Downlink periods is still widely used because of ease of implementation, but fail to accurately model the real measurements. We conclude that our proposed log-normal model clearly tracks our measured results for Uplink and Downlink and can be easily implemented.

ACKNOWLEDGMENT

Scott Fowler was partially supported by EC-FP7 Marie Curie CIG grant, Proposal number: 294182. The work leading to these results has also received funding from the European Unions Seventh Framework Programme (FP7/2007-2013) under grant agreement n^o [609094 RERUM].

REFERENCES

- [1] Cisco. Cisco visual networking index: Global mobile data traffic forecast update, 2013-2018. [ONLINE] http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.pdf, January 2014.
- [2] A. Hanjalic. Shot-boundary detection: unraveled and resolved? *IEEE Transactions on Circuits and Systems for Video Technology*, 12(2):90–105, Feb 2002.
- [3] Shih-Fu Chang, W. Chen, H.J. Meng, H. Sundaram, and Di Zhong. A fully automated content-based video search engine supporting spatiotemporal queries. *Circuits and Systems for Video Technology, IEEE Transactions on*, 8(5):602–615, Sep 1998.
- [4] Xinggong Zhang, Yang Xu, Hao Hu, Yong Liu, Zongming Guo, and Yao Wang. Modeling and analysis of skype video calls: Rate control and video quality. *IEEE Transactions on Multimedia*, 15(6):1446–1457, Oct 2013.
- [5] Skype Inc. Skype features. [ONLINE] Available: <http://www.skype.com>, 2013.
- [6] Chris Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):747–757, Aug 2000.
- [7] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.
- [8] T.K. Moon. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60, Nov 1996.
- [9] T. G. Robertazzi. *Single Queueing Systems, in Computer Networks and Systems: Queueing Theory and Performance Evaluation*. Springer-Verlag, New York, USA, 2000.
- [10] S. Fowler, J. Sarfraz, M.M. Abbas, E. Bergfeldt, and V. Angelakis. Evaluation and prospects from a measurement campaign on real multimedia traffic in lte vs. umts. In *IEEE International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace Electronic Systems (VITAE)*, pages 1–5, May 2014.
- [11] G. W. Corder and D. I. Foreman. *Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach*. John Wiley & Sons, New York, USA, 2009.
- [12] L. Wasserman. *All of nonparametric statistics*. Springer Texts in Statistics, New York, USA, 2007.
- [13] Vilijandas Bagdonavicius, Julius Kruopis, and Mikhail Nikulin. *Nonparametric Tests for Complete Data*. Wiley-ISTE, London, UK, 2011.