Computer Communications 241 (2025) 108244

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Simulating extended reality traffic: An empirical model from user behavior to network packets☆

Luca Mastrandrea, Alessandro Priviero, Gaetano Scarano, Stefania Colonnese *, Tiziana Cattai

Department of Information Engineering, Electronics and Telecommunications, Sapienza University of Rome, Italy

ARTICLE INFO

Keywords:
Extended reality
Traffic model
Hidden Markov Models (HMM)

ABSTRACT

Several components in the design of next-generation networks, including user profiling and network slicing, rely on accurate models of traffic load. In this context, recent studies have focused on various video traffic categories, while traffic associated with extended reality (XR) services has received limited attention. This paper introduces a novel empirical model for 3D XR traffic, developed by encoding real Point Clouds using a standard-compliant codec, and able to account for the dynamic of service sessions and user behaviors over an entire session. Our methodology encompasses multiple temporal scales, ranging from milliseconds to minutes, to account for different phenomena related to both user behavior and encoder settings. Initially, we investigate the packet size distribution at the time scale of a semantic unit, corresponding to the encoding of a single point cloud. We verify that it can be effectively represented by a heavy-tailed Gamma distribution. Then, we illustrate how this insight can be leveraged to model application-layer phenomena. Specifically, we demonstrate the applicability of a general semi-hidden Markov model to capture both the temporal dynamics of service sessions and user behaviors. We provide results in terms of comparison of the empirical and fitting traffic distributions, based on quantile to quantile analysis and statistical tests. We also show how the model can be trained on real data and we provide a pseudo-code demonstrating the model application within a network simulator.

1. Introduction

In next-generation networks [1], traffic models are crucial to profile users, design network slicing and data caching strategies, and analyze system cost-revenues. Advances in multimedia source models the creation of experimental testing environments [2], e.g., for the evolution of multimedia vehicular networks [3]. The literature has investigated video traffic sources for single- and multiview services [4,5], whereas the characterization of the traffic generated by novel eXtended Reality (XR) services is still in its initial stages [6]. XR services in next-generation networks involve the exchange of various types of media, including 360-degree video, and synthetic and natural volumetric data. XR services support a range of multimedia formats and employ different data plane protocols operating under medium- to low-latency constraints, as illustrated in Fig. 1. Among the media transmitted in extended reality services, point cloud data are key to representing virtual and natural 3D objects by sets of points corresponding to the object surfaces. The 3D point coordinates are either captured by active devices such as LiDARs or reconstructed from two or more RGB cameras with different viewpoints, whereas the point colors are obtained by matching the 3D point coordinates with their projections onto one or more RGB camera views. Point clouds, consisting of sets of 3D coordinates associated with color information, can be encoded and transmitted [7] to represent real or virtual objects in three-dimensional space within multimedia services such as XR multi-party calls or XR streaming services. XR services introduce unique challenges for next-generation networks [8], including high throughput demands [7,9] and stringent requirements for low delay and minimal jitter [10]. Therefore, XR traffic needs precise orchestration of network resources, which can benefit of an a priori model of the traffic load dynamics. Modeling XR traffic introduces several challenges [11], due to the significant variability based on the specific application and the type of encoded data, which include natural, synthetic or mixed information resulting in different typical throughputs. While in classical streaming services the traffic depends just on the encoded media quality, possibly adaptive to

☆ This work was supported in part by the European Union - Next Generation EU (EU-NGEU) under the Italian National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.3, CUP B53C22004050001, partnership on “Telecommunications of the Future” (PE00000001 - program “RESTART”), and in part by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU.

* Correspondence to: Via Eudossiana 18, 00184, Rome, Italy.
E-mail address: stefania.colonnese@uniroma1.it (S. Colonnese).

https://doi.org/10.1016/j.comcom.2025.108244
Received 5 January 2025; Received in revised form 17 May 2025; Accepted 3 June 2025
Available online 17 June 2025
0140-3664/© 2025 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

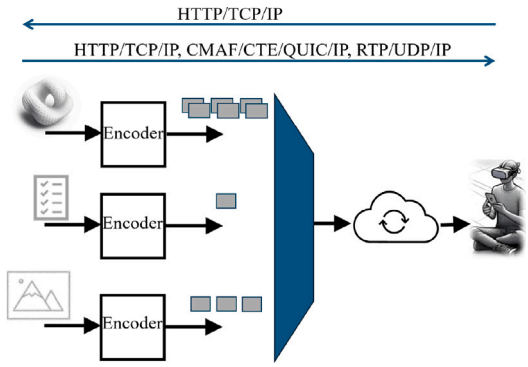


Fig. 1. Extended reality services in next-generation networks envisage the exchange of different kinds of media, under medium to low-latency protocols.

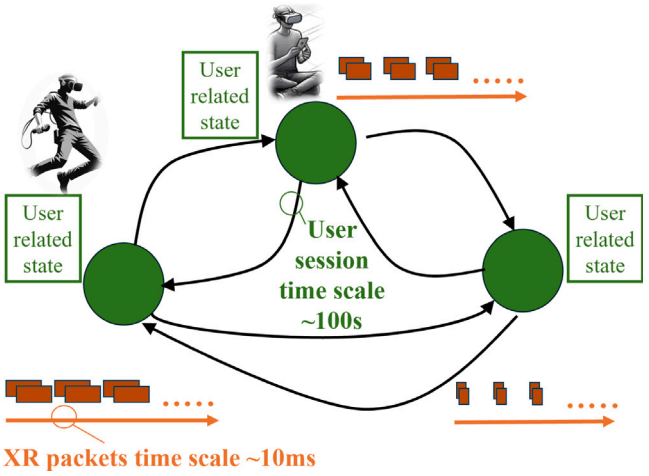


Fig. 2. Schematic representation of the effect of different factors on the XR traffic behavior. This latter is influenced by content formation and encoding but also by user behavior. Those factors operate a different temporal scales, which must be considered for XR traffic modeling.

network quality fluctuations in [12], XR services involve traffic changes related to the user interactions with the immersive environment. For instance, in interactive applications such as gaming the model must account for user states [13] — such as idling, scrolling through an extended reality view, or abruptly changing viewpoints — that generate distinct traffic patterns. Traffic changes during a service session in answer to the user behavior, which induces changes in the statistical features of the transmitted XR data.

Among others, XR data commonly referred to as Point Clouds (PCs) stand out both for relevance and challenging throughput and latency requirements. PC are useful in applications such as training [14], creative production [15], and security or maintenance [16], and necessitate specialized modeling approaches. Existing studies have primarily concentrated on modeling traffic generated by virtual reality applications, particularly gaming services [11,13]. However, a notable gap exists in the modeling of 3D volumetric XR data, especially PCs.

This paper addresses the research gap by introducing an innovative traffic model tailored for XR PC data, accounting for the dependence of XR service traffic on the user behavior and the media encoding settings. Specifically, we present a traffic model for XR volumetric video services, where point cloud data are generated, encoded, and packetized for transmission.

To capture the complexity of XR data sources, the model operates at different time scales associated with XR data generation, which depend on user behavior, network quality of service, and the type of transmitted data. We statistically characterize XR traffic in terms of packet size distribution at temporal scales of tenths of milliseconds. Additionally, it models traffic fluctuations caused by user states at higher time scales, on the order of tenths of seconds. To achieve this, a semi-hidden Markov model is proposed. The model provides a flexible and accurate representation of XR traffic across different time scales, referring to the user's session level, the semantic unit level, and possibly to the encoder output buffer level, as sketched in Fig. 2.

The key contributions of this paper are as follows:

- We characterize the statistical distribution of the size of XR data packets.
- We propose a semi-hidden Markov model to account for rate fluctuation phenomena driven by user behavior, identifying the main parameters and their data driven estimation.
- We outline a traffic source emulator, based on the proposed semi-Hidden MARkov model for XR traffic generation; we refer to this emulator as H-MARX. We also provide examples and guidelines for the use of H-MARX within network simulators designed for XR services.

Preliminary results on the semi-Hidden Markov model appear in [6]. This work extends those results in several directions. Firstly, we expand the statistical analysis of encoded point cloud sequences, and we complement it with a Quantile versus Quantile analysis and statistical tests comparing the empirical distribution of encoded XR packet sizes to the fitted Gamma distribution. The analysis is conducted for different classes of encoded data packets generated by a point cloud encoder [7], specifically referring to the point cloud color, shape and geometry data. Finally, we present the H-MARX emulator including pseudo-code and demonstrating how to apply within a network simulator, where the semi-HMM parameters are tuned using real data.

The paper is organized as follows: Section 2 provides an overview of related work. Section 3 presents the proposed multiscale XR source model. Section 4 empirically characterizes the XR traffic source (i) at the time scale of each encoded point cloud, which represents the single semantic unit, i.e. the single 3D object accessible to the user at a given time instant, and (ii) at larger time scales spanning an entire session. Section 5 shows how the XR traffic model can be used to simulate XR traffic in next-generation networks, introducing the H-MARX traffic source emulator. Finally, Section 6 concludes the paper and discusses future research directions.

2. Related work

The design of advanced multimedia services pushes the need for realistic models of the video traffic, in terms of statistical characterization of the payload offered to the network. The available literature mainly focuses on modeling the dynamics of multimedia source rate adaptation and the user behavior. The dynamics of rate adaptation have been extensively studied, initially for video streaming and more recently for PC traffic [17]. At the application layer, rate adaptation is mainly driven by the estimated download throughput and the state of the receiver's buffer. When the network encounters challenging conditions, such as high latency or the risk of buffer underflow, the application layer typically reduces the transmission rate to minimize disruptions. In contrast, under stable network conditions, the transmission rate is progressively increased to align with the available throughput. This adaptation strategy is effectively modeled using a state-dependent Hidden Markov model. Adjustments to the rate typically occur at regular intervals, often lasting a few seconds, and are influenced by the quality of the communication channel [4]. The probabilities of transitioning between states are determined by the specific rate adaptation algorithm in use [18]. The likelihood of transitioning from one state, representing a particular video rate or quality, can be affected by historical data from lower communication layers, such as retransmission counts or

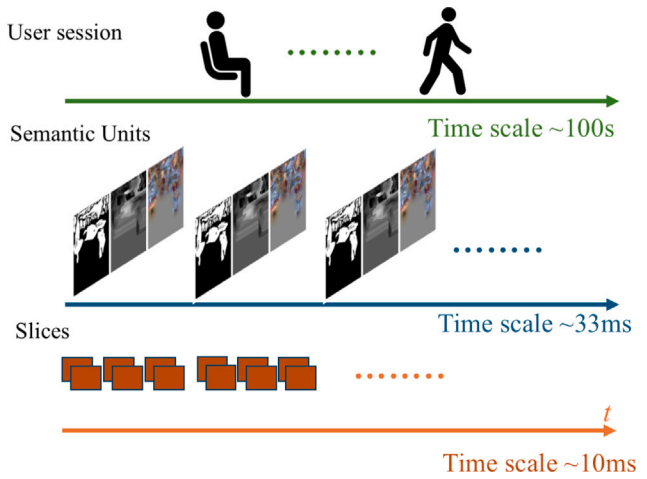


Fig. 3. Graphical representation of multi-scale temporal model. From the up to the bottom of the figure, we have the time scale of session layer, then the layer of the different semantic units, e.g., point clouds, accessible to the user, and finally the slice layer.

latency trends. By incorporating these transitions, the state-dependent semi-Markov model provides a comprehensive representation of how application-layer rate adaptation interacts with evolving network conditions over time. Models of user behavior during a session have been analyzed in different applications, for mobile traffic or web interaction, as described in [19–21]. Concerning video streaming, in [4], the authors address multi-view video streaming and model realistic traffic for diverse scenarios, including rate-adaptation and user behavior model for interactive view selection. In [22] the authors present a comprehensive analysis of mobile-app traffic patterns using Markov models to predict future traffic behavior, whereas Markov chains are introduced in [23] for network traffic prediction at various scales.

The literature on XR spans different kinds of applications, such as mixed reality streaming, 3D scene understanding, showcasing novel acquisition and compression techniques in real-world scenarios. Point clouds are a set of data points lying in the 3D space representing tridimensional shapes or objects. They are widely used in extended reality to represent detailed and accurate 3D models of scenarios [24]. In this context, the quality of point clouds is a crucial aspect to determine their actual usability. Indeed, the study in [25] explores how the PC quality and the viewer's distance from those objects affect user experience in mixed reality environments. Novel quality of experience (QoE) metrics, emphasizing those factors together with subjective perception [26], could be developed. Innovative methods have been introduced [27] to understand and interpret dynamic scenes from 2D images to 3D PCs, focusing on the motion and flow. They propose novel information organization techniques and a new definition of motion, by leveraging deep learning and attention mechanisms in order to improve scene flow estimation in complex environments. Another important challenge is related to efficiently compressing color information in dynamic 3D point clouds. One possible solution identified in [28] is to use affine transformations to reduce the amount of color data necessary to represent a point cloud without visual quality degradation. In [29] an XR offloading traffic model for the case video data enriched by enhanced information.

The XR literature lacks models devoted to XR sources related to dynamic, volumetric content, which are expected to be extremely relevant in novel XR verticals. Differently from state-of-the-art work primarily focused on mixed and augmented reality data from video traffic, we model the specific characteristics of 3D volumetric video data of traffic generated by PC encoding. Specifically, in the following we show that a Hidden Markov model, effectively applied to multi-view Video Coding sources In [5], well catches the main features of PC-based XR traffic.

3. Extended reality source modeling

In this section we present our multi-scale temporal model, illustrated in Fig. 3. In Sub Section 4.1, we begin by analyzing the lower layer, which corresponds to the encoder output, and then in Sub Section 4.2 we describe the higher layer, representing the user session.

3.1. Point cloud encoder characterization

We consider an XR source transmitting 3D volumetric animated content, represented as a sequence of point clouds evolving over time. Each point cloud represents the elementary information unit relevant to the user—namely, the 3D object at a given time instant, referred to as a semantic unit. The temporal resolution of the point cloud sequence ranges from 30 to 60 frames per second (fps), meaning that a semantic unit is generated every 1/60 to 1/30 s, corresponding to a temporal scale on the order of tens of milliseconds. Consequently, the encoder produces output packets at a rate of 30–60 packets per second, or higher if each point cloud is fragmented into multiple packets. We consider a standard-compliant encoder, namely the MPEG V-PCC encoder, which explicitly targets PC video and explores mature coding tools original list designed for 2D data [7]. The coding process is illustrated in Fig. 4. At each time snapshot, the V-PCC encoder projects the point cloud on 2D image planes corresponding to different directions, collecting information on the color of the points, their position and distance from the 2D projection plane. The obtained patches are tailored and juxtaposed into 2D maps for further encoding. Two maps represent, for each 3D vertex, the information on the color and the depth with respect to the projection planes; besides, since the patches do not completely fill the maps, a third map contains the binary information indicating whether the pixel corresponds to an actual point cloud vertex or not. The 2D maps are referred to as the attribute, geometry, and occupancy map. These maps are subsequently encoded using a bidimensional hybrid transform encoder, inspired by the H.265 High Efficiency Video Coding (HEVC) technology [30]. At the encoder output, each 2D attribute, geometry or occupancy frame is encoded into one or more syntactic units named slices. The advantage is that each slice can be independently packetized at the transport layer, making the transmission more robust and suited to mobile environments where transmission errors may either result in rendering errors or delays at the receiver, depending on whether unreliable or reliable transport protocols are adopted causing a degradation of the service.

As in conventional 2D encoding, the frames of each 2D map sequences can be encoded independently from one each other, in the so called INTRA mode (I), or predicted from one or two adjacent frames, using the Predicted (P) or Bidirectionally predicted (B) mode. The periodic introduction of the Intra coding mode reduces the compression efficiency and may introduce latency at the encoder output, but it assures error resilience and random access to the encoded stream. Periodic introduction of INTRA encoded PCs enforces a Group of Picture (GOP) syntactic structure in the encoded bitstream at the output of the hybrid encoder. Each GOP structure starts with Intra encoded data, i.e., data that can be decoded independently without the need of past information. This allows an user to randomly access different part of the encoded sequences. In bidirectional communication applications, such as XR video conference services, the INTRA coding mode is adopted just at the beginning of the sequence or in case of unrecoverable errors, to reduce latency at the encoder and improve encoding efficiency. Finally, there may be applications where the whole sequence of 3D data needs to be accessible independently with respect to the past, for instance in high quality medical applications. A visual description of the three modes appear in Fig. 5.

Hence, the encoding of point cloud XR sequences generates different kinds of packets at the output of the application layer, differing in the represented data (texture color, geometry, and occupancy), and in the coding mode (INTRA or predicted). Besides, as in conventional

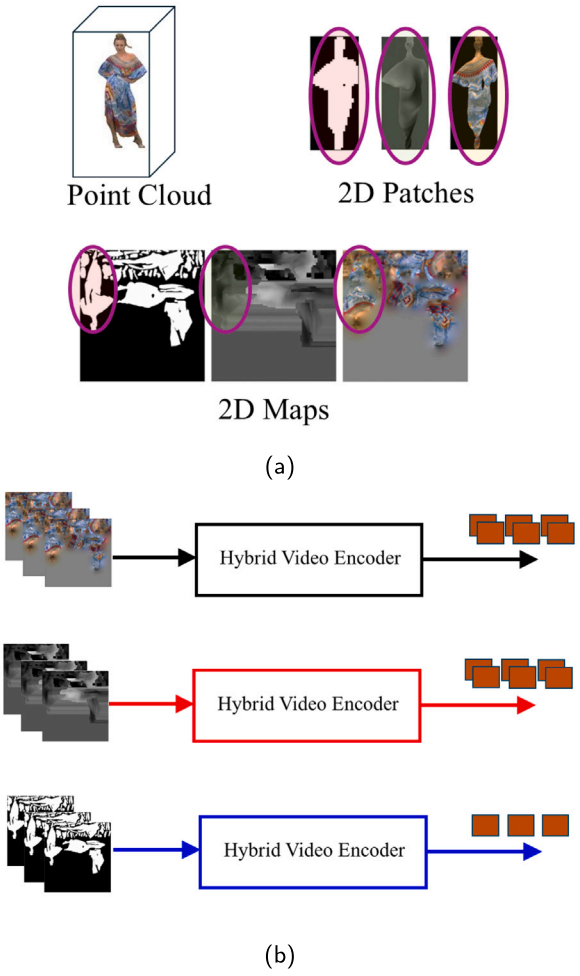


Fig. 4. V-PCC Encoding structure. (a) The V-PCC encoder projects a 3D PC into 2D planes from various directions, and represents the color, position, and distance of points relative to the 2D projection planes by 2D patches. Those patches are arranged to create three distinct 2D maps representing texture, geometry, and occupancy information. (b) The three sequences of maps are encoded using hybrid transform encoding. The encoded data stream is organized in packets called slices, and each frame can be conveyed by one or more slices.

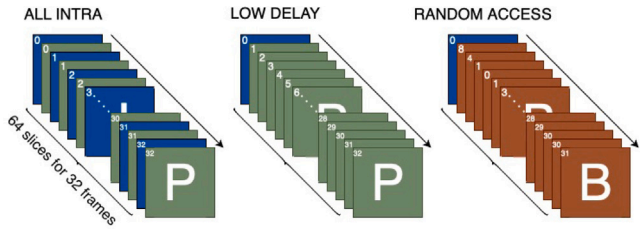


Fig. 5. The slices at the V-PCC encoder output may be independently decodable or organized into a hierarchical dependency structure. The figure shows the slice ordering for the three encoding conditions together with the slice relative frame number. Blue squares represent I-Slices, Green squares P-Slices and Brown squares B-Slices.

2D hybrid encoding, the encoding may be realized at different bitrates corresponding to different visual qualities. Furthermore, the data can be generated within service sessions characterized by different user interactions and requests.

Let us denote as $x_n^{(T)}$, $x_n^{(G)}$, and $x_n^{(O)}$, $n = 0, 1, \ldots \ldots N-1$ the sequences of the sizes of the encoded texture, geometry, and occupancy slices generated at the encoder output. We aim to characterize the cumulative density function:

$$D_{x^{(\cdot)}}(x) = \mathcal{P}\left\{x^{(\cdot)} \leq x\right\} \tag{1}$$

given a set of real encoded traces, and to identify a well fitting distribution. Besides, we will characterize the autocorrelation function of the same sequences, defined as:

$$R_{x^{(\cdot)}}[k] = \mathcal{E}\left\{x_n^{(\cdot)} x_{n+k}^{(\cdot)}\right\} \tag{2}$$

where $\mathcal{E}$ is the expected value. In detail, we will analyze three probability density functions, namely the normal distribution:

$$p_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{3}$$

the truncated Gaussian distribution,

$$p_X(x) = \frac{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)}{D_x\left(\frac{b-\mu}{\sigma}\right) - D_X\left(\frac{a-\mu}{\sigma}\right)}, \quad a \leq x \leq b \tag{4}$$

and the Gamma distribution

$$p_X(x) = \frac{x^{\alpha-1} e^{-x/\theta}}{\theta^\alpha \Gamma(\alpha)}, \quad x > 0,\ \mu = \alpha\theta,\ \sigma^2 = \alpha\theta^2 \tag{5}$$

The truncated Gaussian distribution was proposed for traffic characterization in [10], while the Gamma distribution was shown to accurately describe the video traffic [4,31].

3.2. Hidden Markov model of extended reality traffic

XR services may be involved in a variety of applications, and are in general more interactive and user-dependent with respect to classical multimedia services such as video broadcasting, streaming and conference call. While these services have clear session behavior, for instance the user initiating a streaming and keeping it alive for a given duration, XR communication services may involve a richer session interaction, as for instance in interaction with a virtual shopping assistant or during gaming sessions. The amount of traffic exchanged during the session changes not only in response to bandwidth variations, but also to user behavior. It is well know in fact that in XR services as gaming the users travels through different states throughout a session, e.g. controlling the character, adjusting the character setting, exploring the environment, or just being idle [13]. Therefore, to model and simulate the XR traffic a characterization of both users related and technical features is needed. Fig. 2 illustrates the two-tier model encompassing these different levels. At a higher time scale (on the order of minutes [13]), the user transitions between different states, such as idling or actively navigating the XR environment. Within each state, multiple independently decodable data packets, referred to as Groups of Pictures (GOPs), are generated. Besides, within each state, the GOP bitrate can be selected within a predetermined set of available bitrates R_i, $i = 0, \ldots, N-1$ adapted based on network quality [12]. Each GOP is further composed of individual frames representing different time points, and each frame is encoded using one or more slices.

The proposed model is therefore the two-layer stochastic process illustrated in Fig. 6. Therein, the first (hidden) layer is a discrete-time Markov chain whose states represent different user session states, and the second layer represents the encoded packet size corresponding to a given user state. Further specifications of the model comes from the following observations:

O1 The probability of transitions between user states can be estimated using the Expectation–Maximization algorithm on real traces.

O2 The distribution of the size of the XR packet is state-dependent and its parameters can be estimated by computing the sample mean and variance from real traces.

On top of the already mentioned phenomena at application and session layer, traffic fluctuations can occur as an application layer

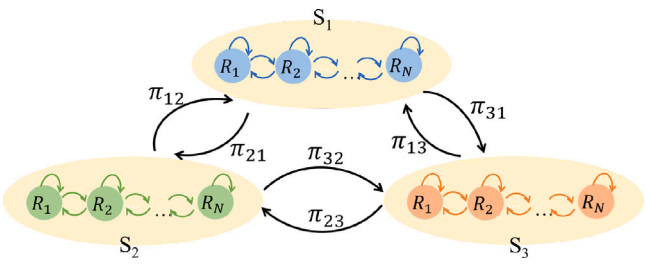


Fig. 6. Two-layer stochastic process adopted to model the XR traffic: the first (hidden) layer is a discrete-time Markov chain whose states represent different user states during the session, and the second layer represents the encoded packet size corresponding to a given user state.

reaction [18] to poor channel conditions, revealed by channel statistics on retransmissions or delays. In streaming services, if low throughput or high latency lead to buffer depletion, the application layer adaptively selects the transmission rate $R_{\bar{i}}$ best suited to the channel throughput; hence, the application layer may select packets encoded at different transmission rates R_i. The rate adaptation process, represented by the state-dependent Markov chains in Fig. 6, is widely adopted in video services [17], and it has been dealt in previous works (see [4] for detailed description). As for XR services, rate adaptation can be carried out according to different criteria, as discussed in [32]. Herein, we instead characterize the traffic evolution at a higher layer, focusing on user modeling and identifying the peculiar distribution of encoded XR traffic packets generated at each user state. Therefore, here we address the characterization of the packets generated at each state, further details about characterization of within-state rate fluctuations can be found in [4].

The user behavior during a session is described as a hidden semi-Markov model, where M states S_m, $m = 0, \dots, M-1$, are visited during the service session for $t \in [0, T-1]$. The state transitions occur at discrete time stamps t_k, where $k = 0, 1, \dots, K-1$. This results in a sequence of states indices $m_{t_k}, k = 0, 1, \dots, K-1$, being visited over time. The duration of each state, $\tau_k = t_{k+1} - t_k$, can follow various probability distributions, allowing the model to accommodate both stationary and non-stationary behavior. The Markov Model is characterized by the probability of transition $\pi_{m_1 m_2}$ between different states S_{m_1}, S_{m_2} $m_1, m_2 = 0, \dots, M-1$. The time interval between state transition t_n is randomly distributed, with a state dependent first order probability density function $p_T(t|S_m), m=0, \dots, M-1$.

The Hidden Markov Model (HMM) describing the user behavior can be parametrized either based on a priori knowledge of user behavior, i.e. by previous analysis of users' habits, as in [19–21], or by fitting available traffic traces, as in [4]. In this latter case, the probability of transition matrix is estimated from a set of observed traces using a suitably formulated version of the Expectation–Maximization (EM) algorithm. This algorithm iteratively estimates the maximum log likelihood function of the unknown HMM parameters and maximizes it, improving the parameters' estimates. The EM algorithm can directly operate on the sequences of packets size representing different snapshots of the encoded PC, computed as $y_n = x_n^{(T)} + x_n^{(G)} + x_n^{(O)}$, $n=0, \dots, N-1$; details of the algorithm are provided in [4]. As an alternative, the sequence of encoded packet sizes y_n is segmented to identify the state transitions $\hat{t}_k$, the segments are classified as belonging to different states, and the matrix of probability of transition is computed from the estimated state sequence.

4. Empirical XR source characterization

In this section, we first empirically characterize the traffic source at the time scale of each encoded semantic unit -i.e. of the point cloud- in sub Section 4.1. These findings will then be used sub Section 4.2 to statistically define the individual states of a higher-layer model that operates on larger time scales. We are therefore able to characterize the traffic load generated by a 3D animated source, focusing on the distribution of the encoded packet size produced by the V-PCC encoder for each input semantic unit (point cloud). The model of the encoded XR traffic is applicable for simulating realistic traffic network design purposes, for instance to optimize resource allocation and orchestration strategies, as further elaborated in Section 5. It is worth noticing that the analysis of the offered traffic load does not model the delays introduced by the communication channel, which depend on the network status and other external factors beyond the XR source. On the contrary, it can be used to develop an XR traffic emulator, to be used to assess the delays into network-oriented simulations, as explained in Section 5.

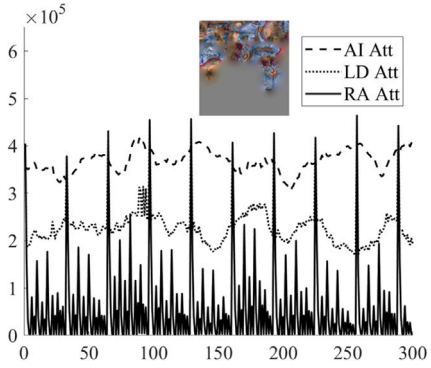
4.1. Empirical distribution of the encoded XR packets

To determine the statistical characteristics of the traffic, we use numerical experiments performed with the V-PCC point cloud encoder developed within the MPEG standardization framework. We study the empirical cumulative distribution function of the real encoded XR traffic packets, and identify the best fitting distribution. Besides, we analyze packet correlation to achieve a compact representation of the source at the semantic unit time scale. We analyze the point cloud (PC) sequences from the dataset [33], with a number of vertices ranging from slightly less than 700,000 to a little over 1,000,000, and acquired at 30 fps. The point clouds are captured at 30 Hz, i.e. at a scale of tenth of milliseconds. The vertices coordinates are quantized over a range of 1024 values, meaning that each vertex is representative of about 1.75 mm for each side, and associated to a 24 bits RGB color attribute information. These sequences are built from multi-camera real-world data and represent humans. Being acquired from the real world and not synthetically generated using computer graphics, this kind of data is highly challenging to encode, while being representative of the XR data to be transferred in different educational, e-health, entertainment and marketing services.

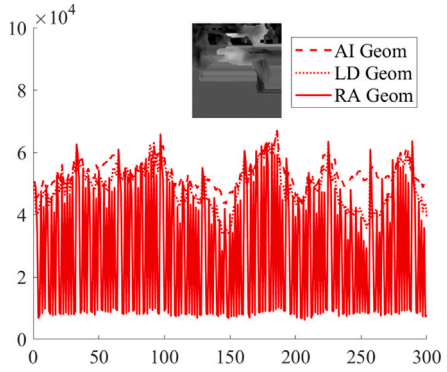
We encoded the PC sequences using the TMC2 testing implementation of the V-PCC encoder [7] which is used for test and benchmarks [34]. For our analyses, we considered three encoding conditions that differ in the frequency of Intra-encoded frames and the ordering of the other predicted frames: All-Intra (AI), where all frames are encoded independently without referencing past frames; Low-Delay (LD) where information is encoded to provide low delay between acquisition and transmission; and Random Access (RA). We provide a visual example of the differences of the traffic generated in the three conditions in Fig. 7. Fig. 7(a), (b) and (c) plot the sequence of slice packet size for attribute, geometry and occupancy maps, versus the point cloud index, for the *Longdress* point cloud sequence, at 30 fps. The time span of the plot corresponds to 10s. For each subfigure, we plot the sequence obtained in the three above mentioned encoding conditions, namely All Intra (AI), Low Delay (LD) and Random Access (RA). From Fig. 7(a), referring to the attribute map, we recognize that the three conditions reflect three different session states, i.e., a state of full activity and interaction, a state of moderate activity, and an idle state periodically providing random access opportunity to resume the active state.

The AI, LD and RA traffic was generated across five different quality rates. The rate control was achieved by adjusting three parameters: the Quantization Parameter (QP) for the Attribute and the Geometry Map, and Precision Parameter (PP), analogous to the QP, for the binary Occupancy Map. The rate control parameters used in the simulations are summarized in Table 1, including the Attribute QP, Geometry QP, and Occupancy PP.

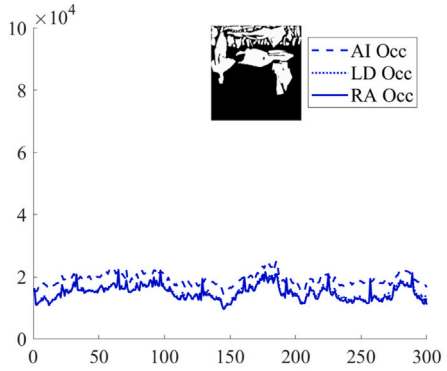
The size of the GOP, i.e., the number of encoded frames within each independently decodable data packet, depends on the kind of considered map and on the selected condition: in the Random-Access condition the GOP size is 32 with one Intra frame followed by 31 bidirectionally predicted frames, in the Low-Delay condition only one



(a) Texture slice size $x_n^{(T)}$[bits], versus the index n



(b) Geometry slice size $x_n^{(G)}$[bits] versus the index n



(c) Occupancy slice size $x_n^{(O)}$[bits], versus the index n

Fig. 7. Sequence of slice packet size [bits] for attribute, geometry and occupancy maps in three encoding conditions, namely All Intra (AI), Low Delay (LD) and Random Access (RA), respectively monitoring three session states, i.e., a state of full activity and interaction, a state of moderate activity, and an idle state (*Longdress* PC, 30 fps. For better visualization, only one out of two slices is represented for the attribute and geometry sequences).

Intra frame is introduced at the beginning of the sequence, while all the remaining are predicted, and in the All-Intra encoding condition every frame is Intra encoded. Therefore, for the Random-Access and Low-Delay conditions we set the GOP size for the texture, geometry, and occupancy data as $N_{\text{GOP}}^{(T)} = N_{\text{GOP}}^{(G)} = N_{\text{GOP}}^{(O)} = 32$, while for the All-Intra $N_{\text{GOP}}^{(T)} = N_{\text{GOP}}^{(G)} = N_{\text{GOP}}^{(O)} = 1$. Each frame of the occupancy map is encoded using one slice, while for the attribute and geometry maps each frame is encoded using two slices. For the All-Intra condition the first slice of each frame is always an Intra slice, and the second slice corresponding to the same frame is always a slice predicted with respect to the first one. In both the Low-Delay and Random-Access condition, only the first of each group of 32 frames is encoded with the first slice being an Intra slice. In these two conditions the occupancy map is encoded using one slice per frame, and only the first of 32 slices is an Intra slice; the remaining are Predicted slices for the Low-Delay case, and Bi-directional predicted slices for the Random-Access case. For the Attribute and Geometry maps, each frame is encoded using two slices, and the first slice is an Intra slice. For the Low-Delay condition, the remaining slices are Predicted following the presentation order of the frames. On the Random-Access condition, the remaining slices are Bi-directional predicted slices starting from the last frame of the GOP, followed by intermediate frames, following the Picture Order Count order: POC = $\{16, 8, 4, 2, 1, 3, 6, 5, 7, 12, 10, 9, 11, 14, 13, 15\}$. Intermediate slices are bidirectionally predicted from adjacent ones, so as to achieve a high compression efficiency while providing random access. These properties are paid in terms of encoding delay, which sums up to the not negligible value of 66 ms at 30 fps.

Table 1
Encoder settings for the Quantization Parameter (QP) of the Attribute and Geometry maps and the Precision Parameter (PP) Occupancy map.

Parameter	*Q1*	*Q2*	*Q3*	*Q4*	*Q5*
Attribute QP	32	28	24	20	16
Geometry QP	42	37	32	27	22
Occupancy PP	4	4	4	4	2

From the experimental results, we consider the size in bits of each frame, corresponding to one Slice in case of occupancy and two slices for the other maps. For each frame, we first analyze separately the Occupancy, Attribute, Geometry frame packets, and then the framewise sum of the three maps. The frame data packets refer to the time scale 1/30 s since all the sequences are acquired at 30 fps.

Fig. 8 illustrates the Quantile versus Quantile (QQ) plot comparing the empirical distribution of the generated XR packet sizes $\left\{x_n^{(T)}, x_n^{(G)}, x_n^{(O)}\right\}$, $n = 0, 1, \ldots \ldots N-1$ and the fitted Gamma distribution, which is the heavy tail distribution commonly used for encoded media. Then, the QQ plots for attribute, geometry and occupancy packets are also shown separated per kind. In all the conditions, the fitting is good, i.e. the data histograms exhibit heavy-tailed distributions. Some discrepancies appear in the attribute map, conveying the texture information, and the geometry map. An interpretation is provided in Fig. 9: when the QQ plot is over the bisectrix of the first sector, as for the attribute map (Att), the sample cumulative distribution function is larger than the Gamma one, and the probability density function exhibit an even heavier distribution i.e., as a rule of thumb large packets appear even more frequently than in a Gaussian distribution with the same variance. Conversely, when the QQ plot is under the bisectrix of the first sector, as for the geometry map (Geo), the sample cumulative distribution function is larger than the Gamma one, and the probability density function exhibit lighter tails, with a smaller probability of very large packets. This can be explained by the fact that geometry information, which relates to the distance of point cloud points from a projection plane, represents the physical geometry of an object and is typically smoother than the texture information, which represents point color.

For comparative purposes, Table 4 reports the results of the Kolmogorov Smirnov test referring to the Gamma distribution and the Truncated Normal Distribution suggested by 3GPP [10] for XR traffic simulation. Fig. 10 presents the QQ plots for the same conditions. Table 4 shows that both the distributions are candidate for XR traffic characterization, but from the QQ plots in Fig. 10 we recognize that the Gamma distribution better captures the behavior of the traffic at the output of a point cloud encoder.

The mean and variance of Slice sizes, along with the parameters (α, β) of the fitted Gamma distributions for different encoding conditions and quality levels, are reported in Tables 2 and 3. These values can be used in a network traffic simulator as explained in the next Section.

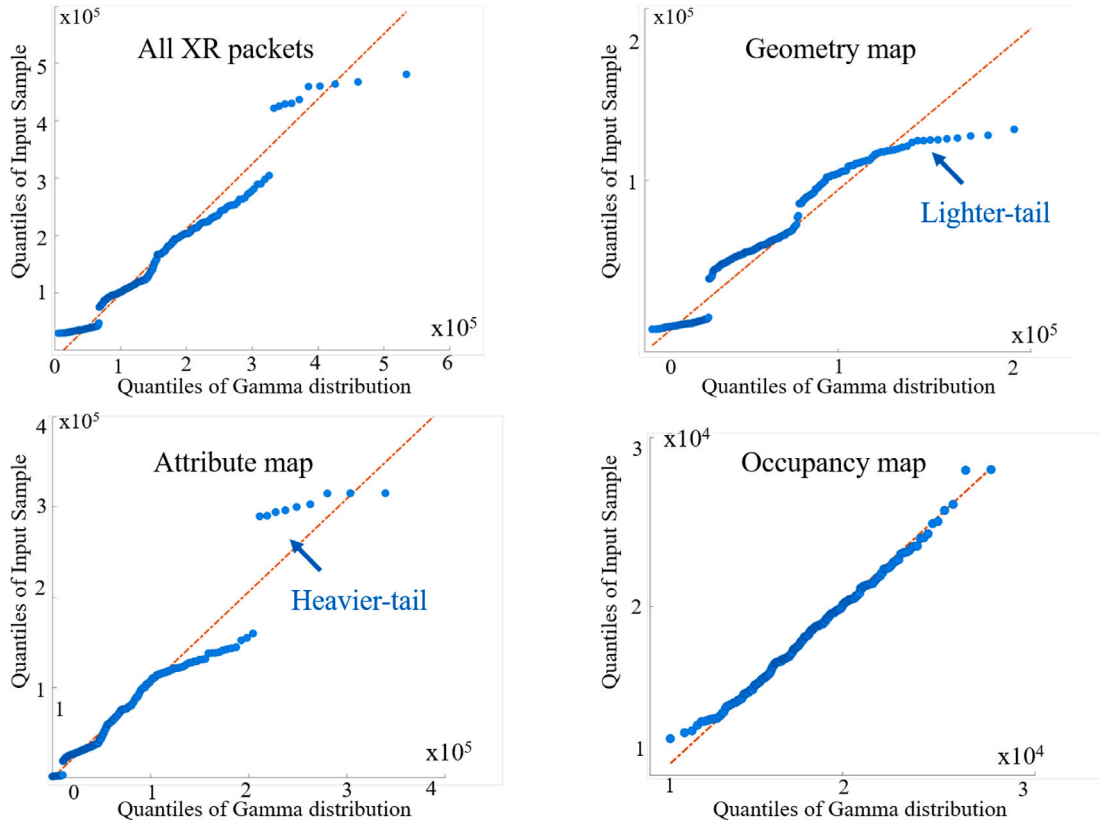


Fig. 8. Quantile versus Quantile (QQ) plot of the empirical distribution of the encoded XR packets and the fitted Gamma distribution. The QQ plot is firstly computed considering all the generated XR packet sizes $\{x_n^{(T)}, x_n^{(G)}, x_n^{(O)}\}$, $n = 0, 1, \ldots \ldots N-1$. Then, the QQ plots for attribute, geometry and occupancy packets are also shown separated per kind.

Table 2
Point cloud Red and black [33]: Slice size mean and standard deviation (μ, σ), and Gamma distribution parameters $(k.\theta)$, for different encoding conditions (All Intra, Random Access, Low delay) and quality levels (Q1–Q5).

Coding mode	Qualities				
All Intra	*Q1*	*Q2*	*Q3*	*Q4*	*Q5*
(μ, σ)	(10 5742, 23.5)	(155 708, 47.0)	(244 614, 15 634)	(418 324, 26 990)	(756 340, 285.9)
(σ/μ)	0.07	0.07	0.06	0.06	0.07
(k, θ)	(193, 549)	(220, 708)	(245, 997)	(241, 1736)	(222, 3406)
Random access	*Q1*	*Q2*	*Q3*	*Q4*	*Q5*
(μ, σ)	(64 304, 37 064)	(90 235, 14.5)	(131 455, 90 951)	(217 578, 47.3)	(434 341, 85.0)
(σ/μ)	0.58	0.62	0.69	0.72	0.63
(k, θ)	(3.43, 18 736)	(2.79, 32 331)	(2.27, 58 000)	(2.18, 99 747)	(2.8, 155 163)
Low delay	*Q1*	*Q2*	*Q3*	*Q4*	*Q5*
(μ, σ)	(81 888, 9.2)	(115 529, 15.0)	(176 624, 28.6)	(313 774, 57.9)	(620 353, 118.8)
(σ/μ)	0.14	0.14	0.15	0.14	0.11
(k, θ)	(51.2, 1600)	(52.3, 2207)	(48.5, 3640)	(57.2, 5488)	(80.9, 7668)

Finally, in Fig. 11 we report the autocorrelation $r_{xx}[k]$ between slices encoded using the low Delay and the Random Access modes. Apart from the slightly different behavior, due to the evident periodicity of the Random Access coding mode, the autocorrelation functions present common characteristics. In particular, we observe that the traffic is rather correlated in time, since the encoding cost is strongly related to the point cloud features (number of points, complexity of the structure) that are typically maintained throughout the session.

In summary, the experiments collected in this subsection demonstrated that the packet sizes are well approximated by a heavy tailed Gamma distribution. This means that packet sizes exhibit significant variability, with some packets being substantially larger due to the highly informative content of point cloud data, which poses challenges for compression. In the next subsection, this result is complemented with the analysis of the state duration.

4.2. Hidden Markov model parameter fitting

To capture the temporal dynamics at a session level, we resort to a Markovian model. This model effectively represents key aspects such as rate adaptation, which occurs on a time scale of seconds, and user behavior, typically observable over durations of ten seconds or more. Specifically, to model user behavior, we resort to a hidden semi-Markov model [5]. The semi-Markov model allows to consider different probability density functions for the random state duration, and it has been successfully applied to multiview video streaming sources. Let S_m, $m = 0, \ldots, M-1$, represent the possible states characterizing user behavior. For instance, in a gaming session they could correspond to the user being idle, zapping, interacting, or other actions. The sequence of states is assumed to follow a hidden semi-Markov process, where transitions between states occur at a pace determined by a random

Table 3
Point cloud Longdress [33]: Slice size mean and standard deviation (μ, σ), and Gamma distribution parameters $(k.\theta)$, for different encoding conditions (All Intra, Random Access, Low delay) and quality levels (Q1–Q5).

Coding mode	Qualities				
All Intra	*Q1*	*Q2*	*Q3*	*Q4*	*Q5*
(μ, σ)	(149 053, 8088)	(260 189, 14 161)	(462 668, 27 074)	(860 134, 52 207)	(1 553 623, 88 713)
(σ/μ)	0.05	0.05	0.06	0.06	0.06
(k, θ)	(0.5465, 51.2)	(0.559, 87.4)	(0.596, 145.6)	(0.6298, 257.2)	(0.6921, 423.7)
Random access	*Q1*	*Q2*	*Q3*	*Q4*	*Q5*
(μ, σ)	(67 285, 51 658)	(106 263, 91 712)	(182 499, 171 935)	(357 033, 319 260)	(830 478, 535 156)
(σ/μ)	0.77	0.86	0.94	0.89	0.64
(k, θ)	(0.7116, 17.8)	(0.7422, 27.2)	(0.7455, 46.5)	(0.7491, 90.7)	(09303, 170.0)
Low delay	*Q1*	*Q2*	*Q3*	*Q4*	*Q5*
(μ, σ)	(94 274, 20 005)	(164 100, 32 473)	(317 123, 51 360)	(662 381, 75 185)	(1 304 103, 108 547)
(σ/μ)	0.21	0.2	0.16	0.11	0.08
(k, θ)	(0.7207, 24.8)	(0.729, 42.8)	(0.7338, 82.1)	(0.7197, 175.0)	(0.7564, 328.1)

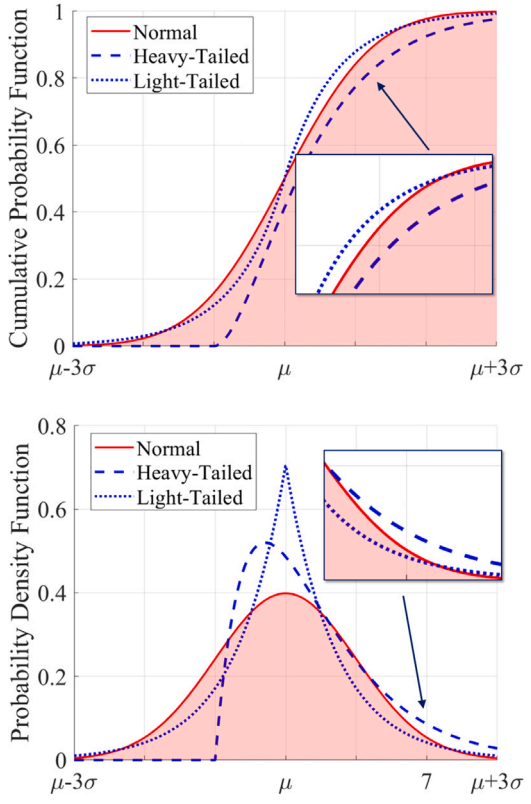


Fig. 9. Interpretation of the QQ plot: the attribute packets have larger cumulative distribution function then the reference one, indicating that their probability density function has heavier tails; the geometry packets have smaller cumulative distribution function then the reference one, indicating that their probability density function has lighter tails.

Table 4
Results of the Kolmogorov Smirnov (KS) test for the fitting of the Gamma distribution over the packet streams on the PC "*Red and black*". Analogous results are obtained for the fitting of the truncated normal distribution in [10].

Coding mode	KS test- Gamma distribution			
	Geo	Occ	Att	All
AI	✓	✓	✓	✓
RA		✓		
LD	✓	✓		✓

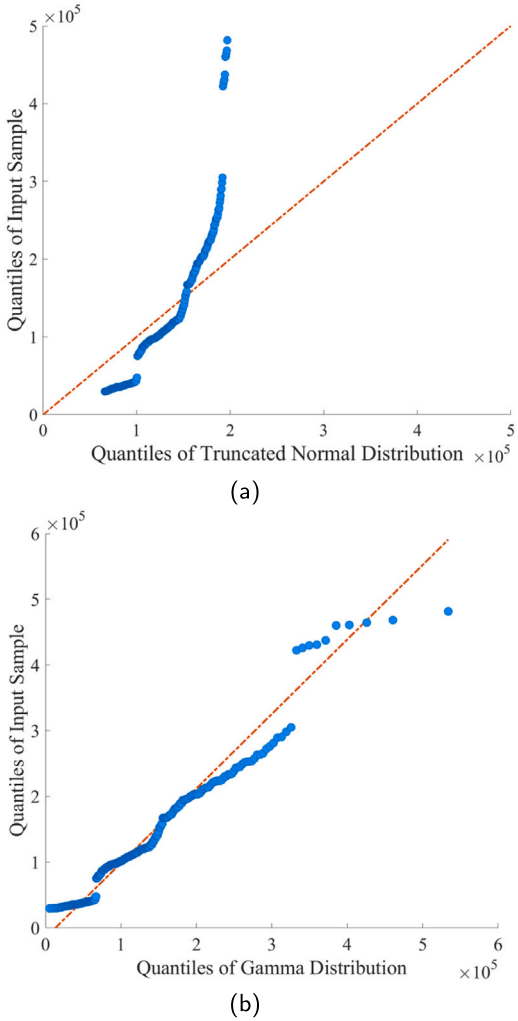
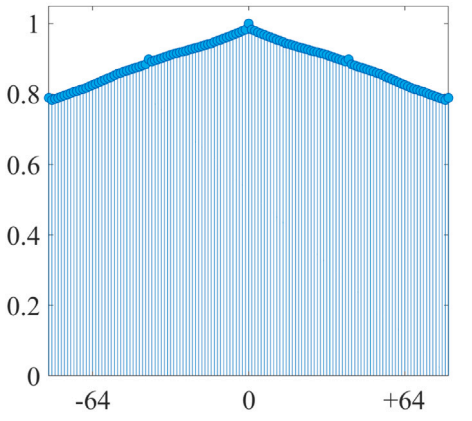
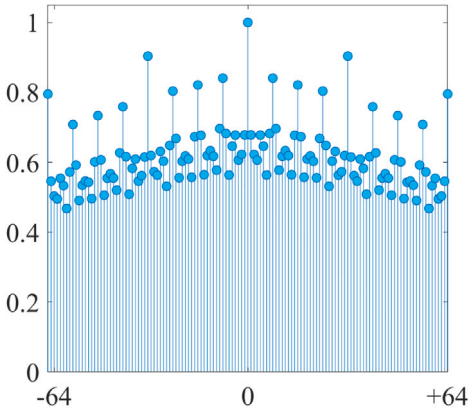


Fig. 10. The Quantile versus Quantile (QQ) plot of the empirical distribution of the encoded XR packets sizes $\{x_n^{(T)}, x_n^{(G)}, x_n^{(O)}\}$, $n = 0, 1, \ldots \ldots N-1$ versus the truncated normal distribution in [10] (a). For comparison sale, we also report the QQ plot referring to the Gamma distribution (b). The Gamma distribution proves to be a much more accurate model of encoded point cloud packets.



(a) $r_{xx}[k]$ versus the index $k=-64,\cdots+64$



(b) $r_{xx}[k]$ versus the index $k=-64,\cdots+64$

Fig. 11. Autocorrelation of the frame packet size for the Low delay (a) and Random access (b) coding modes.

distribution. Specifically, at time t_k, a user transitions from state $S_{m_{t_k}}$ to state $S_{m_{t_{k+1}}}$ with an inter-arrival time $\tau_k = t_{k+1} - t_k$, defined by the characteristics of the specific service.

These transitions encapsulate typical user behavior during an XR service session. For example, a user may alternate between playing, remaining idle, or scrolling through volumetric data. The hidden semi-Markov model effectively captures these dynamics, with the inter-arrival time τ_k approximated using a Poisson distribution. This approach has proven effective for modeling processes such as activity levels in video sequences and user behavior in multiview video services [5].

By incorporating a Markov chain, we can model probabilistic transitions between user states, offering insights into user preferences and interactions. The number of states, M, depends on the service's specific characteristics. For example, in [13], up to five playing modes are identified for gaming users, suggesting that $M = 5$ would be appropriate for modeling their behavior.

The flexibility of the proposed model allows it to be adapted to diverse user profiles, whether human or machine. For instance, in a multi-rig camera surveillance system mounted on a UAV, states such as hovering, charging, or idle could be represented by the Markov chain states, with $M = 3$ in this case. The adaptability of the model enables it to capture a variety of use cases where behavior approximates a Markovian process.

In practical applications, the transition matrix of the Markov chain can be learned from real traffic data corresponding to specific service

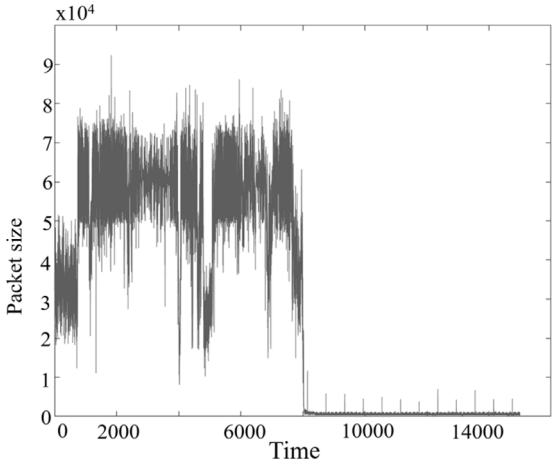


Fig. 12. Example of traffic evolution under a user state change during a cloud gaming session: the packets sizes versus time are extracted from real gaming traffic traces from the dataset in [13].

categories, since the traces present abrupt statistical changes during a user session. This is exemplified in Fig. 12, plotting the encoded packet size with reference to the real cloud gaming traffic traces of the dataset studied in [13]. The fragment refers to a user state change from an active to an idle state during a gaming session.

In [35], the authors outline methods for training such models based on observed user behavior patterns. This ensures that the model remains accurate and aligned with the characteristics of the target application, making it a robust tool for modeling user dynamics in XR and related services.

The Markovian model parameters can be tuned to mimic real sources given an observed sequence of packet sizes. Firstly, an accurate estimate of the sequence of transmitted state can be recovered either directly from the observation of a sequence of packet sizes using the Expectation–Maximization algorithm [5] or by segmentation of the packet size sequence and clusterization of the segments in different states [36]. Once an estimate of the state sequence is available, the Maximum Likelihood estimate $\hat{P}_{ML}$ of the ground truth transition probability matrix P can be obtained. In Fig. 13, we show the results in terms of the Mean Square Error (MSE)

$$\text{MSE} = \|\hat{P}_{ML} - P\|_F$$

where $\|\cdot\|_F$ denotes the Frobenius norm, versus the length of the observed state sequence. We recognize that accurate results are obtained, provided that each state appears in the sequence at least once.

5. The H-MARX traffic simulator

We now detail how the XR traffic model can be used to simulate XR traffic in next-generation networks, introducing the H-MARX traffic source emulator. H-MARX firstly generates the sequence of states describing the user behavior throughout a session, and it randomly draws the state duration; then, the traffic conditional to the state is generated as a series of packets from a heavy tailed Gamma distribution, as suggested by the previous analysis. The detailed steps for network traffic simulation using the H-MARX algorithm are shown in Algo. 0.

The Markov chain parameters related to the user's behavior during the session, including the number of states N, the mean values of the state duration μ_{T_i}, $i = 1,\ldots,M$ can be either a priori set to describe known users behaviors patterns, or estimated by application of the Expectation–Maximization algorithm from short observations of users behavior, as described in [5]. The parameters related to the application layer, i.e., state dependent mean value and variance $\mu_{B_i}, \sigma^2_{B_i}$

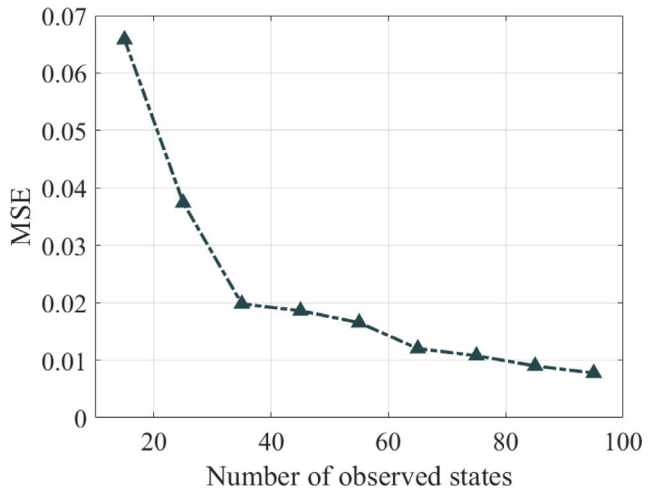


Fig. 13. Mean Square Error of the Maximum Likelihood estimate $\hat{P}_{ML}$ of transition probability matrix evaluated according to the Frobenius norm and plotted versus the number of observed states.

Algorithm 1 H-MARX traffic emulator

Input

- Number of states M
- State transition probability matrix $P \in \mathbb{R}^{M\times M}$
- State-dependent duration means $\mathcal{M}^{(\tau)} = \{\mu_1^{(\tau)}, \dots, \mu_M^{(\tau)}\}$
- State-dependent packet size means $\mathcal{M}^{(y)} = \{\mu_1^{(y)}, \dots, \mu_M^{(y)}\}$
- State-dependent packet size standard deviations $\mathcal{S}^{(y)} = \{\sigma_1^{(y)}, \dots, \sigma_M^{(y)}\}$

for each packet index $t = 1, 2, \dots$ **do**

State selection: Generate the state at time t, $S[t] \in \{S_1, \cdots S_M\}$, given the previous state $S[t-1]$ and the transition probability matrix P.

State duration sampling: Draw a Poissonian random variable $T[t]$ with mean value $\mu_{T[t]} \in \mathcal{M}^{(\tau)}$ dependent on the current state $S[t]$.

XR packet size generation: Draw a set of $T[t]$ Gamma distributed XR packets sizes $\mathbf{y}[t] = [y_{n_1}, \cdots y_{n_{T[t]}}]$, with state-dependent mean value $\mu_{\mathbf{y}[t]}$ and variance $\sigma^2_{\mathbf{y}[t]}$.

end for

The H-MARX traffic emulator can be inserted in a network simulation framework: each XR user exhibits a unique session behavior pattern, modeled using the semi-HMM model process. State transitions occur at time intervals governed by a Poisson distribution, with average durations on the scale of minutes. Within each state, the packet sizes are sampled from a Gamma-distributed random variable, with parameters determined by the specific state and quality level. These assumptions lead to the generation of synthetic traffic providing a concrete framework for realistic simulation of the application layer in network simulations.

For instance, the H-MARX simulator enables the emulation of heterogeneous user behaviors by allowing the configuration of distinct parameter sets for each user profile. A representative application of this capability is the simulation of extended reality (XR) traffic, which facilitates the design and evaluation of network slicing mechanisms for differentiated service provisioning. By systematically varying user profile parameters, the simulator can be employed to assess their impact on the trade-offs between resource allocation and system performance. These findings can further support the development of service charging models based on user-specific resource demands and quality of service requirements.

To sum up, H-MARX can be used for different analysis, such as: (i) determining bandwidth and throughput requirements for supporting XR applications in advanced networks; (ii) profiling users by analyzing system latency under varying conditions; (iii) accurately simulating the specific demands of XR applications, such as high data rates, minimal latency, and support for a large number of users in both physical and virtual settings; (iv) enabling the creation of a network Digital Twin, where the model is trained to replicate XR user load and quality of experience, thereby enhancing synchronization between physical and digital systems. (v) orchestration and recovery mechanisms for applications such as streaming, group training, patrolling to mention a few.

H-MARX can be adapted and extended by changing the following parameters:

- number of the hidden Markov model states;
- constant or random state duration, and state duration probability density function (e.g. Poissonian);
- state dependent transition probability matrix;
- state dependent probability density function (PDF) of the encoded semantic unit packet size.

In the following, we show an example of how H-MARX is used a simulated environment. Firstly, we consider a Markovian source, oscillating between different states in accordance to the transition matrix mimicking the behavior of a real source. An example, corresponding to the following transition matrix:

$$P = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.4 & 0.3 & 0.3 \\ 0.2 & 0.2 & 0.6 \end{bmatrix}$$

is depicted in Fig. 14.

An example of how H-MARX can be applied to generate a realistic source of XR traffic is as follows. Firstly, a sequence of states is generated in accordance to the model as in Fig. 14. Based on different studies in the literature [19–21], the time interval between state transition is assumed to be Poisson distributed, with state dependent mean. Hence, the state duration is drawn from the selected Poissonian distributions. Fig. 15 shows the histogram of a sample random sequence of state duration t_n observed throughout a state sequence; the three state dependent duration histogram peaks are recognized. Throughout the state sequence, a set of state dependent chunk traces are then generated, as in Fig. 16. Therein, the x-axis represents the index of the state sequence; in turn, each state has its own random duration, during which a variable number of packets are generated, at a rate of $1/\tau$ seconds. The y-axis represents the packet sizes.

For example, H-MARX can be used to simulate traffic load for a gaming user scenario, with three user states representing user's pausing, zapping, scrolling actions. This requires defining a synthetic traffic generator structured with three states, such as Idle, Scroll, and Zapping.

Finally, H-MARX can be extended by introducing several quality levels within each state, to model rate adaptation strategies in reaction to network performances. The research about rate adaptation in XR services is still open, and we leave this relevant development for further study.

6. Conclusion and future work

This paper provides an empirical XR traffic model useful for next generation network design purposes. We propose a two-tier model: the first stage addresses coarse-grain traffic fluctuations at different time scales, ranging from several minutes for user behavior, to seconds for rate adaptation; the second stage relates to fine-grain fluctuations of tenths of milliseconds of the sole encoded XR data packet. The first stage encompasses a hidden (semi) Markov model of the traffic fluctuations while the second stage resorts to a conditional heavy tailed

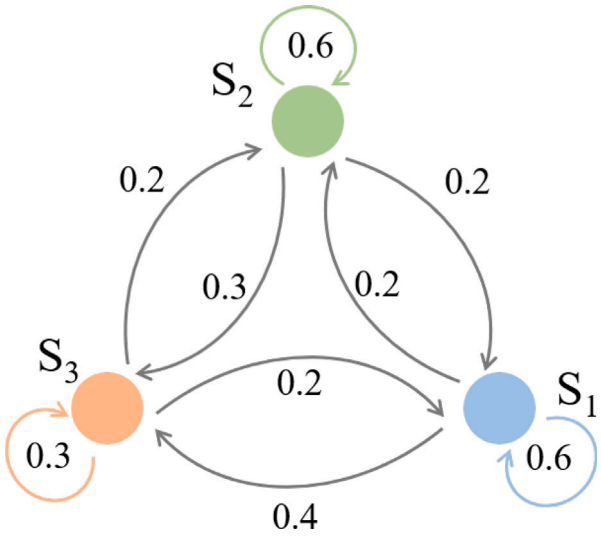


Fig. 14. Graph representation of the semi-HMM model exploited by the H-MARX algorithm. In this simulation, S_1 corresponds to Idle, S_2 to play(scroll) and S_3 to play (zap).

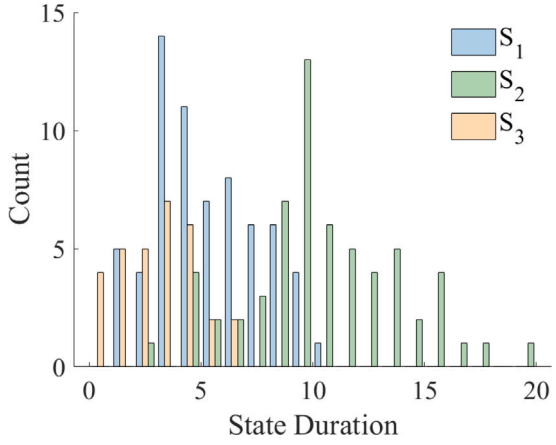


Fig. 15. Histogram of the state dependent durations, generated by the H-MARX algorithm throughout a sequence of 330 states.

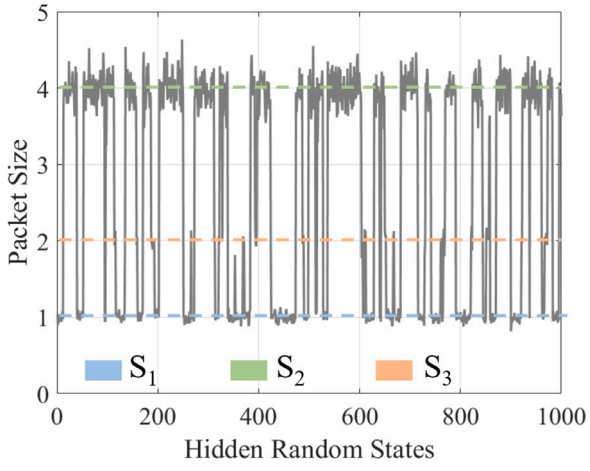


Fig. 16. Size of packets generated by the H-MARX algorithm throughout a series of hidden randomly drawn states: the average value corresponding to each state is shown by colored horizontal bars.

data distribution, found by numerical studies on real encoded data. The empirical model is driven by few parameters and it can be adopted as a tool in network design, for the purposes of XR oriented network slicing, orchestration or cost-revenue analysis in next generation networks. Further work will integrate the proposed model to develop artificial intelligence aided resource orchestration techniques, leveraging the model to generate several realistic data for training purposes.

CRediT authorship contribution statement

Luca Mastrandrea: Writing – review & editing, Writing – original draft, Software, Investigation. **Alessandro Priviero:** Writing – review & editing, Writing – original draft, Software, Investigation. **Gaetano Scarano:** Writing – review & editing, Methodology, Investigation, Conceptualization. **Stefania Colonnese:** Writing – review & editing, Supervision, Methodology, Investigation, Conceptualization. **Tiziana Cattai:** Writing – review & editing, Methodology, Investigation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the European Union - Next Generation EU (EU - NGEU) under the Italian National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.3, CUP B53C22004050001, partnership on “Telecommunications of the Future” (PE00000001 - program “RESTART”) and in part by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU.

Data availability

Data will be made available on request.

References

[1] J. Navarro-Ortiz, P. Romero-Diaz, S. Sendra, P. Ameigeiras, J.J. Ramos-Munoz, J.M. Lopez-Soler, A survey on 5G usage scenarios and traffic models, IEEE Commun. Surv. Tutor. 22 (2) (2020) 905–929.
[2] S. González, W. Castellanos, P. Guzmán, P. Arce, J.C. Guerri, Simulation and experimental testbed for adaptive video streaming in ad hoc networks, Ad Hoc Netw. 52 (2016) 89–105.
[3] X. Jiang, F.R. Yu, T. Song, V.C. Leung, Resource allocation of video streaming over vehicular networks: A survey, some research issues and challenges, IEEE Trans. Intell. Transp. Syst. 23 (7) (2021) 5955–5975.
[4] S. Colonnese, P. Frossard, S. Rinauro, L. Rossi, G. Scarano, Joint source and sending rate modeling in adaptive video streaming, Signal Process., Image Commun. 28 (5) (2013) 403–416.
[5] L. Rossi, J. Chakareski, P. Frossard, S. Colonnese, A poisson hidden Markov model for multiview video traffic, IEEE/ACM Trans. Netw. 23 (2) (2014) 547–558.
[6] T. Cattai, L. Mastrandrea, A. Priviero, G. Scarano, S. Colonnese, Glancing at extended reality: An empirical model of 3D animated XR data traffic, in: 2024 IFIP Networking Conference, IFIP Networking, IEEE, 2024, pp. 68–76.
[7] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, A. Tabatabai, An overview of ongoing point cloud compression standardization activities: Video-based (v-PCC) and geometry-based (G-PCC), APSIPA Trans. Signal Inf. Process. 9 (2020) e13.
[8] Z. Liu, Q. Li, X. Chen, C. Wu, S. Ishihara, J. Li, Y. Ji, Point cloud video streaming: Challenges and solutions, IEEE Netw. 35 (5) (2021) 202–209.
[9] M. Quach, J. Pang, D. Tian, G. Valenzise, F. Dufaux, Survey on deep learning-based point cloud compression, Front. Signal Process. 2 (2022) 846972.
[10] 3GPP, Technical Specification Group Services and System, Technical Report TR 26.928, 2020, V1.2.1, Extended Reality (XR) in 5G.
[11] M. Lecci, M. Drago, A. Zanella, M. Zorzi, An open framework for analyzing and modeling XR network traffic, IEEE Access 9 (2021) 129782–129795.

[12] M. Hosseini, C. Timmerer, Dynamic adaptive point cloud streaming, in: Proceedings of the 23rd Packet Video Workshop, 2018, pp. 25–30.
[13] M. Carrascosa, B. Bellalta, Cloud-gaming: Analysis of google stadia traffic, Comput. Commun. 188 (2022) 99–116.
[14] C.L. Taborda, H. Nguyen, P. Bourdot, Engagement and attention in XR for learning: Literature review, in: International Conference on Virtual Reality and Mixed Reality, Springer, 2025, pp. 207–242.
[15] G.W. Young, N. O'Dwyer, A. Smolic, Volumetric video as a novel medium for creative storytelling, in: Immersive Video Technologies, Elsevier, 2023, pp. 591–607.
[16] Y. Wang, Z. Su, N. Zhang, R. Xing, D. Liu, T.H. Luan, X. Shen, A survey on metaverse: Fundamentals, security, and privacy, IEEE Commun. Surv. Tutor. 25 (1) (2022) 319–352.
[17] J. van der Hooft, H. Amirpour, M.T. Vega, Y. Sanchez, R. Schatz, T. Schierl, C. Timmerer, A tutorial on immersive video delivery: From omnidirectional video to holography, IEEE Commun. Surv. Tutor. 25 (2) (2023) 1336–1375.
[18] J. Kua, G. Armitage, P. Branch, A survey of rate adaptation techniques for dynamic adaptive streaming over HTTP, IEEE Commun. Surv. Tutor. 19 (3) (2017) 1842–1866.
[19] I. Tsompanidis, A.H. Zahran, C.J. Sreenan, Mobile network traffic: A user behaviour model, in: 2014 7th IFIP Wireless and Mobile Networking Conference, WMNC, IEEE, 2014, pp. 1–8.
[20] H. Zhang, W. Ni, X. Li, Y. Yang, Modeling the heterogeneous duration of user interest in time-dependent recommendation: A hidden semi-Markov approach, IEEE Trans. Syst. Man Cybern.: Syst. 48 (2) (2016) 177–194.
[21] A. Bianco, G. Mardente, M. Mellia, M. Munafò, L. Muscariello, Web user-session inference by means of clustering techniques, IEEE/ACM Trans. Netw. 17 (2) (2008) 405–416.
[22] G. Aceto, G. Bovenzi, D. Ciuonzo, A. Montieri, V. Persico, A. Pescapé, Characterization and prediction of mobile-app traffic using Markov modeling, IEEE Trans. Netw. Serv. Manag. 18 (1) (2021) 907–925.
[23] I. Guarino, A. Nascita, G. Aceto, A. Pescapà, Mobile network traffic prediction using high order Markov chains trained at multiple granularity, in: 2021 IEEE 6th International Forum on Research and Technology for Society and Industry, RTSI, IEEE, 2021, pp. 394–399.
[24] F. Wirth, J. Quehl, J. Ota, C. Stiller, Pointatme: efficient 3d point cloud labeling in virtual reality, in: 2019 IEEE Intelligent Vehicles Symposium, IV, IEEE, 2019, pp. 1693–1698.
[25] M. Nguyen, S. Vats, S. Van Damme, J. Van Der Hooft, M.T. Vega, T. Wauters, C. Timmerer, H. Hellwagner, Impact of quality and distance on the perception of point clouds in mixed reality, in: 2023 15th International Conference on Quality of Multimedia Experience, QoMEX, IEEE, 2023, pp. 87–90.
[26] M. Nguyen, S. Vats, S. Van Damme, J. Van der Hooft, M.T. Vega, T. Wauters, F. De Turck, C. Timmerer, H. Hellwagner, Characterization of the quality of experience and immersion of point cloud video sequences through a subjective study, IEEE Access (2023).
[27] X. Xiang, R. Abdein, W. Li, A. El Saddik, Deep scene flow learning: From 2D images to 3D point clouds, IEEE Trans. Pattern Anal. Mach. Intell. (2023).
[28] C. Cao, M. Preda, T. Zaharia, Affine transformation-based color compression for dynamic 3D point clouds, in: 2022 IEEE International Conference on Image Processing, ICIP, IEEE, 2022, pp. 1556–1560.
[29] D.G. Morín, D. Medda, A. Iossifides, P. Chatzimisios, A.G. Armada, A. Villegas, P. Peréz, An extended reality offloading IP traffic dataset and models, IEEE Trans. Mob. Comput. (2023).
[30] G.J. Sullivan, J.-R. Ohm, W.-J. Han, T. Wiegand, Overview of the high efficiency video coding (HEVC) standard, IEEE Trans. Circuits Syst. Video Technol. 22 (12) (2012) 1649–1668.
[31] I. Rubin, S. Colonnese, F. Cuomo, F. Calanca, T. Melodia, Mobile HTTP-based streaming using flexible LTE base station control, in: 2015 IEEE 16th International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM, IEEE, 2015, pp. 1–9.
[32] H. Wang, H. Dong, A. El Saddik, Immersive multimedia communication: State-of-the-art on extended reality streaming, ACM Trans. Multimed. Comput. Commun. Appl. (2025).
[33] M. Krivokuca, P.A. Chou, P. Savill, 8I voxelized surface light field (8iVSLF) dataset, 2018, ISO/IEC JTC1/ SC29/WG11 MPEG, Input Document M42914.
[34] Video codec based point cloud compression (V-PCC) test model. https://github.com/MPEGGroup/mpeg-pcc-tmc2. (Accessed June 2023).
[35] S. Sun, S. Li, Y. Li, B. Moran, W.S. Rowe, Smartphone user tracking by incorporating user orientation using a double-layer HMM, IEEE Trans. Veh. Technol. 71 (7) (2022) 7780–7790.
[36] G. Budel, F. Frasincar, D. Boekestijn, DBHC: Discrete Bayesian HMM clustering, Int. J. Mach. Learn. Cybern. (2024) 1–16.