computers

MDPI

Article

Estimation of 5G Core and RAN End-to-End Delay through Gaussian Mixture Models

Diyar Fadhil [1,2,†] and Rodolfo Oliveira [1,2,*,†]

[1] Departamento de Engenharia Electrotécnica e de Computadores, Faculdade de Ciências e Tecnologia, FCT, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
[2] Instituto de Telecomunicacoes, 1049-001 Lisbon, Portugal
* Correspondence: rado@fct.unl.pt
† These authors contributed equally to this work.

Abstract: Network analytics provide a comprehensive picture of the network's Quality of Service (QoS), including the End-to-End (E2E) delay. In this paper, we characterize the Core and the Radio Access Network (RAN) E2E delay of 5G networks with the Standalone (SA) and Non-Standalone (NSA) topologies when a single known Probability Density Function (PDF) is not suitable to model its distribution. To this end, multiple PDFs, denominated as components, are combined in a Gaussian Mixture Model (GMM) to represent the distribution of the E2E delay. The accuracy and computation time of the GMM is evaluated for a different number of components and a number of samples. The results presented in the paper are based on a dataset of E2E delay values sampled from both SA and NSA 5G networks. Finally, we show that the GMM can be adopted to estimate a high diversity of E2E delay patterns found in 5G networks and its computation time can be adequate for a large range of applications.

Keywords: end-to-end delay; quality of service; Gaussian mixture model; cellular networks

Citation: Fadhil, D.; Oliveira, R. Estimation of 5G Core and RAN End-to-End Delay through Gaussian Mixture Models. *Computers* **2022**, *11*, 184. https://doi.org/10.3390/computers11120184

Academic Editor: Paolo Bellavista

Received: 1 November 2022
Accepted: 6 December 2022
Published: 12 December 2022

1. Introduction

In recent years, the number of smart devices and applications has increased exponentially, so it is predicted that by 2025 the number of devices connected to the network will reach 30 billion [1]. This growth means that users' expectations regarding security, real-time operation and privacy protection should be considered in addition to properly executing tasks or monitoring systems. The wide range of applications requires diverse network characteristics, from physical and transmission technologies to routing and transport protocols capable of supporting different service sets.

Network analytics lets network operators explore various practical models to troubleshoot configuration issues, enhance network efficiency, cut operational costs, identify potential security threats, and plan the development of the network. For instance, Nokia has proposed the Network Data Analytics Function (NWDAF) [2], a network analytics engine capable of analyzing parameters in different circumstances for performance optimization and capacity planning, credential misuse, and cloud security. The Quality-of-Service (QoS) metrics are usually seen as performance indicators of the network status. The network operators employed different methods to enhance user experiences by improving QoS metrics. For instance, providing a robust connection with less delay for Device to Device (D2D) communication is guaranteed by resource allocation and power control in 5G cellular networks [3]. One of the most tangible QoS metrics directly related to user experiences and qualification is the End-to-End (E2E) delay. The E2E delay is the time needed to transfer a packet from one endpoint to another, i.e., the time between the instant the transmission starts at the source node and the instant the packet is ultimately received at the destination node.

From a network management viewpoint, it is essential to identify the network E2E delay profile so that its suitability for supporting different delay-constrained services can be assessed over time. For instance, in 5G networks, Ultra-Reliable Low Latency Communication (URLLC) applications demand low latency networks [4], while other kinds of applications, such as opportunistic sensing, do not have such requirements. Due to the different requirements of each service in terms of throughput, reliability, and time sensitivity, it is crucial to know the E2E delay probabilistic features. Using probabilistic models to determine the E2E delay distribution is crucial to support different delay management strategies [5].

Different methodologies have been proposed to model the E2E delay. Queuing theory-based (QT) models were proposed to compute the mean E2E delay by considering the stochastic properties of the queue's arrival and departure random processes [6–8]. Although QT models are very popular in the literature for modeling and predicting delay, these are too dependent on the statistics of the queue's arrival and departure random processes, which can easily change over time due to the variety of applications and generated traffic patterns. Additionally, QT models only allow the computation of expected values (e.g., expected queue delay or expected delay), not providing any insight into the distribution of the delay. Network Calculus-based (NC) models were also proposed in the literature [9–11]. NC models allow the computation of a delay bound of a flow traversing the network [12]. Although delay bounds are useful for time-sensitive scenarios, such as the parameterization of timeout values, they do not provide a high level of description as the one obtained from the estimation of its distribution.

In a nutshell, QT models provide long-term management based on the arrival and service random processes and only allow the computation of expected values (e.g., expected queue delay or expected delay). However, they do not provide any insight into the distribution of the delay. NC models allow the control of unforeseen events and problems by determining the worst cases events and readiness for solving these issues. However, similar to QT models, NC models do not characterize the distribution of the delay. On the contrary, the model proposed in our work focuses on a more detailed estimation of the E2E delay, where instead of estimating a bound or an expected value of the delay, the goal is on estimating the distribution of the delay.

When characterizing the E2E delay through various distribution models, the critical challenge is determining which distributions represent the experimental data collected over time. Due to the network's heterogeneous nature, mainly due to the diversity of different radio management policies and network Core technologies available in 5G networks [13], the characterization of the E2E delay involves the parameterization of different delay patterns that might change significantly over time. Consequently, the E2E delay often does not follow a single known Probability Density Function (PDF) but a mixture of them. This diversity motivates a modeling approach based on probabilistic mixture models that combine two or more distributions to increase the model's accuracy. To this end, the research question to be addressed in this work is centered on how the distribution of the E2E delay can be accurately estimated in a short amount of time. The scientific hypothesis explored in this work aims at evaluating the feasibility of estimating the E2E through a Gaussian Mixture Model (GMM) [14]. The main question to be answered is how the multiple parameters of the GMM, such as the number of GMM components and the number of samples adopted in the estimation process influence the estimation accuracy and its computation time. An open problem in GMM is the selection of the methodology that should be used to obtain its optimal parameters for a given number of components. In Section 2, we provide a literature review of different methodologies capable of computing the GMM parameters and its pros and cons are also identified.

The innovative aspects of this paper include the following contributions:

- The identification of a methodology to estimate the distribution of the E2E delay based on 5G data obtained over time. The GMM is adopted to estimate the PDF of the E2E delay of 5G networks, considering both standalone and non-standalone operation

and different network subsystems such as the Radio Access Network (RAN) or the Core network;
- The influence of the number of GMM components and number of data samples on the estimation accuracy;
- The evaluation of the GMM's computation time as a function of the number of model components as well as the number of samples used as input;
- An assessment of GMM's accuracy versus its computation time, which allows the characterization of the tradeoff between both features.

The rest of this paper is organized as follows: Section 2 introduces the literature review on the parameterization of GMM. The estimation methodology is presented in Section 3. Section 4 introduces the 5G dataset and the different scenarios considered in the experiments evaluated in this work. Section 5 evaluates the estimation performance for the different experiment scenarios. Finally, Section 6 concludes the paper.

Regarding the notation adopted in this work, we use $Pr(X = x)$ or simply $Pr(x)$ to represent the probability of X. Vectors are represented in upper case, upright boldface type, e.g., $\mathbf{X}$.

2. Literature Review

A GMM is defined as a parametric probability density function that consists of a linear combination of multiple Gaussian distributions [15]. Different approaches to estimate the distributions' parameters and weights values based on observed data include the Maximum Likelihood Estimation (MLE), Expectation-Maximization (EM) [16], Minimum Message Length (MML), Moment Matching (MM), and Penalized Maximum Likelihood Expectation-Maximization (PML-EM) [14]. The MLE approach maximizes the likelihood function between a known distribution and the observed data. The MML method is an information measurement for statistical comparison. The MM method finds the unknown parameters by obtaining the expected values of the random variables' powers of the population distribution model equal to the sample moments [17]. MM can be employed as an alternative approach for MLE in most complex problems due to its simple, easy, and fast computation. The PML-EM is an approach to estimate the parameters in cases when the likelihood is relatively flat, which makes MLE estimation determination difficult. The EM is an iterative algorithm that maximizes the likelihood expectation between data and a mixture of distributions. However, EM as convergence rate is influenced by the initialization random values, and it is hard to define the number of the distributions adopted in the mixture model and how they affect the accuracy of the approximation. EM's dependency on the initialized values is one of the main causes of slow convergence, as indicated in [18].

The GMM has received significant attention in the literature, particularly to support the estimation of QoS network parameters. The work in [19] investigates how to estimate the link-delay distributions based on end-to-end multicast measurements and adopting an MLE-based GMM model. In [20], a known conditional distribution and an unknown finite Gaussian mixture is proposed to approximate the weighting of the GMM components, showing that higher accuracy is achieved when compared to the EM algorithm. The EM algorithm also has drawbacks. Generally, the EM algorithm faced three main issues: First, determining the initial value may change the results and the converging state and rate. Second, it is hard to define the number of mixture model distributions and how they affect the accuracy of the approximation. Finally, the convergence rate is prolonged in some cases and may take a long time to achieve an accurate solution.

The majority of the research works addressing the improvement of the EM algorithm emphasize that EM dependency on the initialized value is its main drawback. Several research works explore different approaches to solve the random initialization of the different GMM variables. The work in [18] proposed a robust EM clustering algorithm for GMM in two phases and formulated a new method to solve the initialization problems in EM. In a further step, this work also proposes a scheme to automatically obtain an optimal

number of clusters. The proposed approach is evaluated using experimental examples to demonstrate the outperformance of conventional EM algorithms. The work in [21] proposes an innovative approach to estimate the mean of the distributions of a mixture model based on local maximum, and the numerical results presented in the paper reveal that the proposed algorithm achieves higher performance than the Naive EM algorithm.

The number of mixed components is one of the most critical issues in GMM. The work in [22] proposed an improved EM algorithm to select the number of components of GMM and simultaneously estimate the weight of these components and unknown parameters. The evaluation results illustrate a better performance in estimating the distribution parameters and consistency in determining the number of components. Moreover, ref. [23] suggested a new method to estimate the GMM components and to identify the jitter cause. The EM algorithm is adopted to determine the best match parameters to fit the observations. The authors also consider the Bayesian Information Criterion (BIC) algorithm to determine the number of GMM distributions and eliminate the initial value selection problem for the EM algorithm.

As a widely used approach to obtain the MLE function, the EM algorithm convergence speed can be a critical issue. There is a wide range of investigations focused on this issue. For instance, in [24] the authors adopted the Anderson acceleration technique. The evaluation results with different simulations and numerical examples show that the number of iterations to obtain convergence is reduced by applying Anderson acceleration on the EM algorithm. Moreover, the work in [24] proposed a new method to address the EM algorithm issue with multimodal likelihood functions. In these cases, the EM algorithm may get trapped into a local maximum, resulting in long convergence cycles to reach the optimal solution. The method proposed in [25] attempts to optimize the initial random values to avoid local maximum points, targeting the maximization step. Simulation results confirm that optimizing the initial value improves the convergence rate and avoids a local trap. The GMM model has been adopted in several scenarios. For instance, ref. [26] suggests a new clutter elimination method based on GMM and EM estimation, which attempts to estimate and perform fast clutter with a small amount of data. The work in [27] provides a comprehensive analysis of actual latency values collected among various data center locations. The work suggests a new GMM approximation based on the simple box approximation algorithm for the round-trip time distribution. The GMM approximation can then be used to simulate and emulate the deployment of applications and services in the cloud.

The works cited so far aim at regenerating the collected data based on all their information. On the contrary, in the methodology followed in this work we determine the GMM parameters that better fit the sampled data. The accuracy of the estimation model is characterized as a function of the number of GMM components and a variable number of input samples. Our goal is to compare the accuracy of the proposed estimation methodology with all empirical data contained in the dataset. Additionally, the GMM estimation time is also studied to assess the feasibility of using the proposed methodology in real-time applications and services.

3. Estimation Methodology

This section systematically analyzes the required data from datasets and explains the estimation methodology. The first part describes how a 5G dataset is filtered and handled to compute the E2E delay of the both Core and RAN subnetworks. In the second part, the EM algorithm is introduced and a methodology to estimate the E2E delay and determine the GMM parameters is described.

3.1. System Model

The system model block diagram is illustrated in Figure 1.

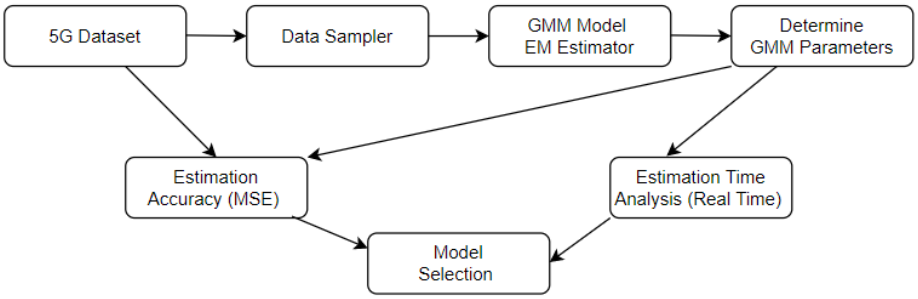

Figure 1. Methodology followed in the estimation process.

In this work we adopt the 5G Campus dataset [28] as a source of 5G E2E delay samples. The E2E delay equals the time it takes to send a packet traveling from the source node to the destination. The Wireshark software is adopted to capture all received and transmitted packets in a specific Network Interface Card (NIC). To determine the exact E2E delay, the timestamp of the sent packet is subtracted from the timestamp of the received one. Moreover, the NICs of the measuring devices capture the network status of the 5G network (Non-Standalone (NSA)/Standalone (SA) architecture) in both download and upload operation modes, and for various packet sizes and packet generation rates. The E2E delay is gathered for 5G's Core and RAN. The E2E delay experienced by the packets in the 5G Core networks is mainly due to the encapsulation and decapsulation of both downlink and uplink communications. The Core E2E delay refers to the period of time an upload or download packet traverses the Core network, including all the processing tasks carried out by the different servers belonging to the Core network. The RAN E2E delay is experienced by the packets transversing the RAN and is measured independently from the Core E2E delay. The RAN E2E delay can be measured in both upload and download directions, i.e., from the user to the Core or from the Core to the user, respectively. A detailed introduction of the 5G Campus dataset [28] is introduced in the next section.

The data contained in the dataset is then used by a "Data Sampler", which forms consecutive sets $\mathbf{X} = \{x_1, x_2, \cdots, x_T\}$ of T samples that are used as the input of the estimation process. The sampler can decide on what kind of data to select from the dataset, including the type of 5G architecture (SA/NSA), the upload or download types of operation, the E2E delay sampling period regulated by the period of probing packets transmission, and the size of the probing packets that vary the network's load. The sets of T samples are then passed to the "GMM Model EM Estimator" to compute the parameters of the GMM model through the EM algorithm. The output of the estimation process is then evaluated by considering the consecutive sets of samples. The evaluation of the estimation accuracy, represented by the block "Estimation Accuracy (MSE)" is based on the Mean Squared Error (MSE) of the empirical data and estimated distributions. The estimation time is also evaluated and takes into account the computation time needed to determine the GMM parameters for a specific set of samples and is measured in the computer implementing the estimation methodology.

3.2. Estimation Process

Although Gaussian distributions can model an impressive number of probabilistic scenarios, certain phenomena follow unknown distributions demanding more complex modeling approaches. The mixture models can be a possible solution for these cases. Given that any natural process may depend on several independent factors that form several subpopulations, the GMM models the subpopulations that can be mixed to describe the whole distribution of the population.

A GMM is defined as a parametric PDF consisting of a linear combination of multiple Gaussian PDFs. The mixture models are usually used for multimodal or multi-peak PDF data. Fitting the multimodal data with a single distribution usually leads to very low

accuracy. The mixture models are used to combine different distributions that better match the data probability density. The Gaussian distribution is adopted in the mixture models due to its theoretical and computational benefits to represent massive datasets [15]. Each Gaussian component is used to represent the subpopulations within an overall population. Three primary parameters define each component of a GMM: the mean, the standard deviation, and the weight. The Gaussian mixture model can be represented as follows

$$Pr(x) = \sum_{i=1}^{K} w_i \mathcal{N}(x \mid \mu_i \sigma_i), \tag{1}$$

where x is the data sample, w_i is the mixture weight for the i-th component, with $i = 1, \ldots, K$, and K is the number of mixed Gaussian distributions, aka GMM components. Each GMM component is defined as follows (a Gaussian PDF)

$$\mathcal{N}(x \mid \mu_i, \sigma_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{(x - \mu_i)^2}{2\sigma_i^2} \right), \tag{2}$$

with mean μ_i and standard deviation σ_i. The mixture weights satisfy the constraint

$$\sum_{i=1}^{K} w_i = 1. \tag{3}$$

Therefore, the GMM parameters vector is represented by $\mathbf{P}$ as follows

$$\mathbf{P} = \{w_1, \mu_1, \sigma_1, \cdots, w_K, \mu_K, \sigma_K, \}. \tag{4}$$

The parameters $\mathbf{P}$ are estimated based on the samples. The MLE is a well-known method that aims to find the mixture model parameters and maximize the GMM likelihood function. We assume that the vector of the samples is represented by $\mathbf{X} = \{x_1, \ldots, x_T\}$, and the samples $x_1, \ldots, x_T$, are independent. Therefore, the likelihood function is written as

$$Pr(\mathbf{X} \mid \mathbf{P}) = \prod_{t=1}^{T} p(\mathrm{x}_t \mid \mathbf{P}), \tag{5}$$

where T represents the number of samples' data. The optimization of the non-linear function of parameters $\mathbf{P}$ in an MLE problem has no closed-form [26]. A practical solution for the MLE of the parameters $\mathbf{P}$ can be determined by a numerical approach such as the iterative EM or similar ones [20]. Although different approaches to parametrize the GMM parameters may provide closed-form expressions [20], its computation complexity is severely increased by the number of distributions used in the mixture model. For instance, if we assume that the GMM consists of $K = 7$ components, the MM estimation needs to calculate and solve 21 equations of the moments to determine the estimations of the parameters. The EM algorithm is a well-known approach often adopted to estimate the GMM parameters due to its iterative behavior and improved computational performance.

In EM, initial random values are assigned to $\mathbf{P}$, which are used to determine subsequent estimation values iteratively. In the first step, the initial values of all parameters are determined and employed as an input of the iterative EM algorithm to compute subsequent values of the different parameters. The initial values of the parameters in $\mathbf{P}$ at the time instant 0, denoted as $\mathbf{P}^{(0)}$, are computed as follows

$$\mu_i^{(0)} = \frac{T * rand}{2} \sum_{t=1}^{T} x_t, \tag{6}$$

$$\sigma_i^{(0)} = \left(\frac{1}{T} \sum_{t=1}^{T} (x_t - \bar{x})^2 \right)^{\frac{1}{2}}, \tag{7}$$

$$w_i^{(0)} = \frac{1}{K}, \tag{8}$$

where *rand* represents a random number sampled from a uniform distribution between zero and one and $\bar{x}$ represents the average of the samples in $\mathbf{X}$. An iterative cycle is then started until the estimated parameters $\mathbf{P}$ reach a specific convergence threshold. The EM algorithm relies on an iterative approach divided into two steps:

1. **E-step**: In the Expectation step, the expectation of the likelihood function is calculated based on the observed data in $\mathbf{X}$ and the current model parameters at time instant m denoted by $\mathbf{P}^{(m)}$.
2. **M-step**: In the Maximization step, the expectation of the likelihood function is used to compute new model parameters $\mathbf{P}^{(m+1)}$ that maximize the conditional distribution given by the samples in $\mathbf{X}$ and the parameters $\mathbf{P}^{(m)}$. The symbols m, and $m+1$ indicate consecutive iterations. In the E-step, $\mathbf{P}^{(m)}$ is used to indicate the current model parameters. In the M-step, $\mathbf{P}^{(m)}$ is used to determine the subsequent model parameter $\mathbf{P}^{(m+1)}$. Expanding the E-step and taking separate derivatives concerning the different parameters (M-step), we obtain the equations as follows

$$w_i^{(m+1)} = \frac{1}{T} \sum_{t=1}^{T} \Pr\left(i \mid x_t, \mathbf{P}^{(m)}\right), \tag{9}$$

$$\mu_i^{(m+1)} = \frac{\sum_{t=1}^{T} \Pr\left(i \mid x_t, \mathbf{P}^{(m)}\right) x_t}{\sum_{t=1}^{T} \Pr\left(i \mid x_t, \mathbf{P}^{(m)}\right)}, \tag{10}$$

$$\sigma_i^{(m+1)} = \sqrt{\frac{\sum_{t=1}^{T} \Pr\left(i \mid x_t, \mathbf{P}^{(m)}\right) x_t^2}{\sum_{t=1}^{T} \Pr\left(i \mid x_t, \mathbf{P}^{(m)}\right)} - \hat{\mu}_i{}^2}, \tag{11}$$

where

$$\Pr\left(i \mid x_t, \mathbf{P}^{(m)}\right) = \frac{w_i^{(m)} \mathcal{N}\left(x_t \mid \mu_i^{(m)}, \sigma_i^{(m)}\right)}{\sum_{k=1}^{K} w_k^{(m)} \mathcal{N}\left(x_t \mid \mu_k^{(m)}, \sigma_k^{(m)}\right)}, \tag{12}$$

The subsequent GMM estimation parameters vector is given by

$$\mathbf{P}^{(m+1)} = \{w_i^{(m+1)}, \mu_i^{(m+1)}, \sigma_i^{(m+1)}\}, \quad i = 1, \ldots, K. \tag{13}$$

In the EM algorithm, we adopt a stop condition based on a convergence threshold and a maximum number of iterations. The EM algorithm stops when the difference parameter D_m is lower than the convergence threshold γ. D_m is given by

$$\begin{aligned} D_m = \sum_{k=1}^{K} (&|\mu_k^{(m+1)} - \mu_k^{(m)}| + |\sigma_k^{(m+1)} - \sigma_k^{(m)}| \\ &+ |w_k^{(m+1)} - w_k^{(m)}|). \end{aligned} \tag{14}$$

When the threshold condition $D_m < \gamma$ is not met, the EM algorithm is stopped for $m = 25,000$ iterations.

4. Evaluation Methodology

This section presents the evaluation methodology in detail. The 5G dataset, testbed, and E2E delay measurement methodology are described, as well as the scenarios considered in this work (NSA/SA/Upload/Download/packets size, and packet rate). At the end of the section, we present the key performance indicators and different experiments are defined to be assessed in Section 5.

4.1. 5G Dataset

One of the main goals of 5G technology is to provide high-speed access and low-latency communications. The 5G Campus is a dataset obtained with 5G networks implementing NSA or SA operation modes [28]. Figure 2 illustrates the 5G NSA and SA blocks and the difference between these topologies in terms of data and control plane. The 5G NSA network uses 5G New Radio (5G NR) to enhance the current 4G LTE network and Evolved Packet Core (EPC) to access the Internet. In this operation mode, the LTE eNodeB sends the control plane data to connect the end device, while the 5G NR and 4G enodeB provide dual connectivity techniques for data planes. A 5G SA leaves the 4G radio and Core behind and utilizes the 5G NR and 5G Core (5GC) to transfer data and control. A 5G NSA is proposed as a step towards switching from 4G to 5G cellular technology.

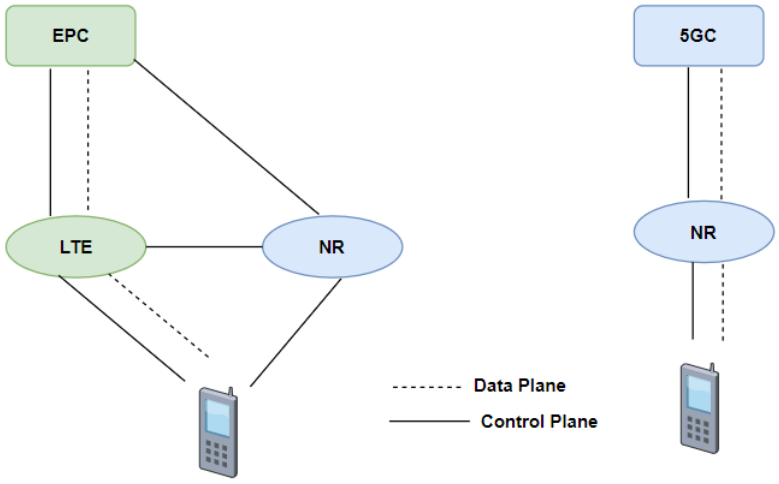

Figure 2. The 5G NSA and SA topologies and identification of the data and control planes.

Specifically, the 5G Campus dataset includes One Way Delay (OWD) within Core and end devices via the RAN in SA and NSA modes and considering both upload and download traffic directions. The testbed for collecting data is presented in [28] and consists of different devices implementing various scenarios and gathering their OWD information through nanosecond precision measurements. The 5G Campus dataset collects data from the testbed, which employs the 5G NR and 4G packet Core for implementing a 5G Core block in both SA and NSA scenarios. The RAN network includes both RAN baseband units, and antennas are separately used by 5G and 4G networks. The end devices are capable of receiving both SA and NSA traffic. There is a traffic generator device to generate and timestamp the packets to measure end-to-end delay.

Testbed

The 5G networks integrate different components, including the Core, the RAN, and the end devices, which are responsible for dedicated duties. The Core block prepares the

generated data for transmission via the RAN network. In other words, the Core network is responsible for encapsulating and decapsulating data and managing the routing and transmitting process. The RAN network provides media for the Core network to reach end devices. The RAN network contains a RAN Baseband Unit (RAN BBU) and RAN antennas. The RAN BBU modifies the digital signals to the analog ones for the RAN antenna to spread in the wireless coverage area. The end devices receive the signals with their antenna and convert them to digital data.

In the 5G Campus testbed [28], in addition to mentioned devices, there are two devices to manage data traffic and capture the E2E delay. A traffic generator with two NICs generates traffic and timestamps transmitted and received packets with nanosecond precision. Moreover, a switch provides connection services within the traffic generator, network Core, and RAN. In addition, to mentioned RAN and Core devices for both 5G NR and 4G LTE networks, the end devices are connected directly to the traffic generator to add timestamps and collect related data. The testbed implements both SA and NSA network topologies and is capable of collecting download and upload OWD measurements for both RAN and Core elements.

There are two kinds of E2E delay measurements in the dataset, which indicate the delay experienced by a packet traveling from the end device to the traffic generator and vice versa. The RAN E2E delay only includes the delay experienced by a packet when traversing the RAN. The Core delay includes the delay experienced in the Core, mainly due to packet encapsulation and decapsulation. The sum of both RAN and Core delays represents the total amount of delay experienced by a packet in the 5G network.

As shown in Figure 3, the RAN download scenario is explained with red arrows and their corresponding step number. In the RAN download scenario, the data generator creates and timestamps a packet with TS_{RD1}, and forwards it to the switch, redirecting the received data to the Core network to encapsulate and send it back to the switch. The switch sends the packet to RAN BBU and antennas via wired interfaces. The end devices receive the transmitted data via the cellular wireless network and forward the received data to the traffic generator after decapsulation via wire interface to add a timestamp TS_{RD2} to the received packet. The difference between the received and sent packet timestamp TS_{RD2}-TS_{RD1} determines the OWD for download. It should be mentioned that the 4G RAN is used for data and control plane in addition to the 5G RAN if the network mode is NSA; otherwise, only the 5G RAN is used.

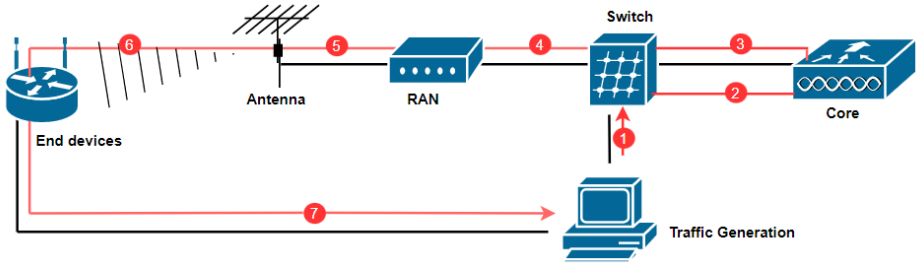

Figure 3. The RAN download testbed and steps to measure the OWD.

The previous process is performed in the reverse direction to find out the upload delay. As shown in Figure 4, the RAN upload scenario is explained with green arrows and their corresponding step number. The GPRS Tunneling Protocol (GTP) encapsulates and decapsulates transferred data in the upload process. In the RAN upload scenario, the data generator creates the packet with timestamp TS_{RU1} and forwards it to the end devices. The end devices encapsulate the received data and propagate it via its antennas. On the other side, the RAN antennas receive the data and send it to the RAN BBU, then reach the switch. The Core network receives the packet from the switch, decapsulates it, and forwards it back to the switch. In the end, the switch sends the decapsulated packet to the data generator,

which stamps it with the timestamp TS_{RU2}. OWD can be easily obtained by subtracting the transmitted timestamp from the received one, TS_{RU2}-TS_{RU1}.

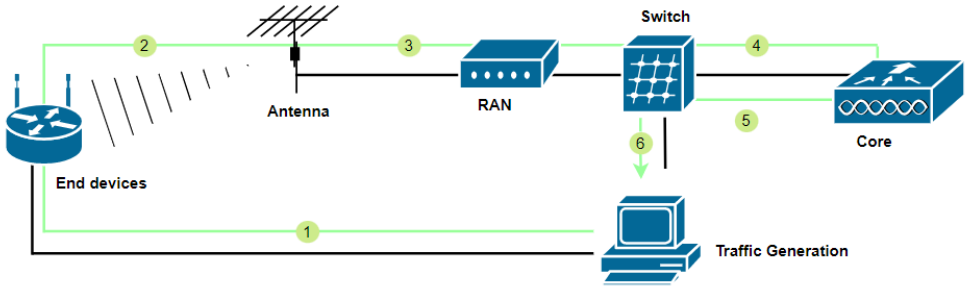

Figure 4. The RAN upload testbed and steps involved to measure the OWD.

As shown in Figure 5, the Core download scenario is explained with red arrows and their corresponding step number. In the Core download scenario, the data generator creates a packet and timestamps it with TS_{CD1}, and then forwards it to the switch, redirecting the received data to the Core network to encapsulate and send it back to the switch. The switch sends the packet to RAN BBU and simultaneously mirrors this packet to the data generator to determine the delay in the Core network. The data generator timestamps the received packet with the stamp TS_{CD2}. The differential between the mirrored copy of the Core network packet and the sent timestamp, TS_{CD2}-TS_{CD1}, determines the OWD for Core download.

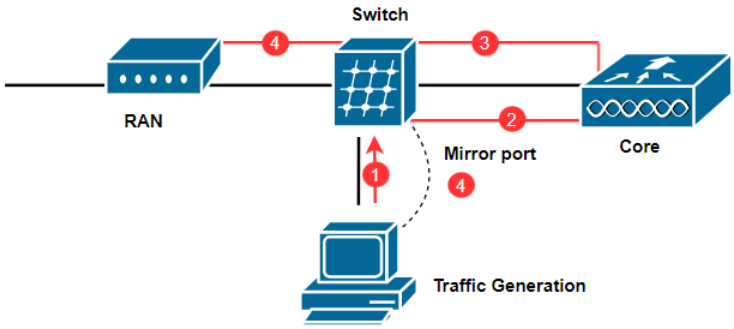

Figure 5. The Core download testbed and steps to measure the OWD.

A similar process is carried out in the reverse direction to find out the upload delay. As shown in Figure 6, the Core upload scenario is explained with green arrows and their corresponding step number. In the Core upload scenario, the switch mirrors the received packet from RAN when it forwards it to the Core network for decapsulation. The data generator timestamps the packet with TS_{CU1}. On the other hand, the Core network decapsulates the packet and transmits it back to the switch, which forwards it to the data generator. The data generator timestamps the packet traversing the Core with the stamp TS_{CU2}. The OWD is computed by TS_{CU2}-TS_{CD1}. Finally, we highlight that in the 5G Campus dataset the same device, i.e., the Traffic Generator, is always used for stamping the packets, avoiding the need for clocks' synchronization as proposed in [29,30].

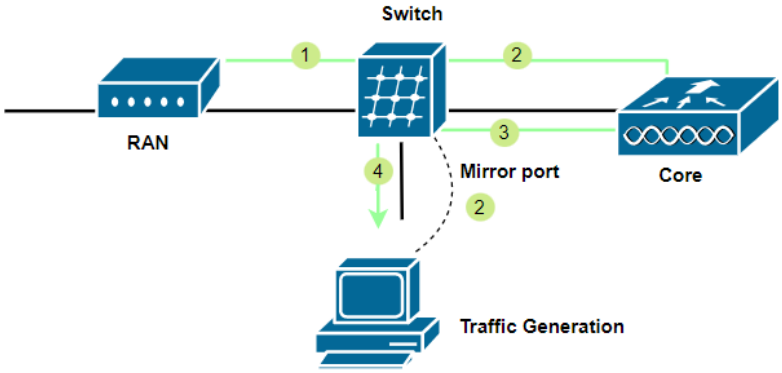

Figure 6. The Core upload testbed and steps required to compute the OWD.

The packets received and transmitted in the NICs are captured with the Wireshark software package. In addition, the dataset collects the packet size and the packet rate. The dataset contains single batches of data exchanged during 1000 s having into account the following features:

1. Network topology (SA/NSA);
2. Delay measurements (RAN/Core);
3. The stream direction (Download/Upload);
4. The packet size (128/256/512/1024/2048 bytes);
5. The packet rate (10/100/1000/10,000/100,000 packets per second).

4.2. Experiments

The 5G Campus dataset provides about 200 sub-datasets based on network topology, delay measurements, stream directions, packet size, and packet rates. Among all these scenarios, the sub-datasets with more subpopulation and less out-of-range data are selected for the evaluation proposed approach. Although the proposed approach can accurately trace the PDF for all datasets, the performance evaluation is characterized for the four scenarios in Table 1.

Table 1. Scenarios of the 5G dataset adopted in the performance evaluation of the E2E delay estimation methodology.

Scenarios	Topology	Delay Type	Stream Direction	Packet Size [bytes]	Packet Rate [packets/s]
Scenario 1	NSA	Core	Download	1024	10
Scenario 2	NSA	Core	Upload	256	10
Scenario 3	SA	RAN	Download	1024	100
Scenario 4	SA	RAN	Upload	128	100

The scenarios in Table 1 include different network topologies, delay at specific subsystems, stream directions, packet sizes, and packet rates. Due to different data rates and constant data collection periods of 1000 s in the 5G Campus dataset, the number of collected packets, and consequently E2E delay samples, is given by $No_{Data} = 1000 \times PacketRate$.

In further analysis, we have used No_{Data} samples to determine the accuracy of the model as a function of the number of GMM components, for $K = 2, 3, 5, 8, 10, 12$. For $PacketRate = 10$ packets/s, a dataset contains 10,000 packets, and considering the sample rate $1/100, 1/50, 1/20, 1/10, 1/5, 1/2$, and 1, we obtain subsets of samples with $T = 100, 200, 500, 1000, 2000, 5000$, and 10,000 samples, respectively.

5. Performance Results

This section presents the simulation results and evaluates the performance of the method described in Section 3. The MSE is used to find out how the PDF estimated with the GMM is close to the empirical PDF of the dataset and is defined as follows

$$MSE = \frac{1}{N}\sum_{k=1}^{N}(y_k - \hat{y_k})^2, \tag{15}$$

where N represents the number of discrete points of the PDF, $\hat{y_k}$ represents the GMM PDF value of the discrete point k, and y_k represents the value of the PDF of the empirical data.

The four selected scenarios in Table 1 have been employed to determine the model's accuracy as a function of the number of GMM components and the number of samples. The threshold γ was set to 1×10^{-6}.

As mentioned before, the number of iterations to reach convergence is one of the essential metrics for evaluating the EM algorithm performance. When the initial values randomly chosen are more accurate, the time to reach a certain level of convergence decreases. Table 2 summarizes the number of iterations of the EM algorithm and the MSE achieved for the different number of components in each scenario. The EM algorithm is computed for datasets with sample rate 1. Based on the numerical results, when the number of components increases, the number of iterations required to reach the convergence threshold increases because more parameters need to be estimated. It is worth nothing that the limiting value of 25,000 iterations was never reached for $\gamma = 1 \times 10^{-6}$. On the other hand, the MSE decreases with the number of components because a higher number of components leads to a higher number of degrees of freedom to model the data.

Table 2. Number of iterations to compute the GMM and MSE achieved with the model.

Scenarios	Number of Components	Number of EM Iterations	MSE
Scenario 1	2	39	1.25×10^{-5}
	3	86	8.86×10^{-6}
	5	347	5.49×10^{-6}
	8	1853	4.24×10^{-6}
	10	2488	3.72×10^{-6}
	12	4901	2.61×10^{-6}
Scenario 2	2	32	2.97×10^{-5}
	3	67	1.74×10^{-6}
	5	256	4.55×10^{-6}
	8	1215	4.04×10^{-6}
	10	1729	1.11×10^{-6}
	12	3243	8.23×10^{-7}
Scenario 3	2	42	1.73×10^{-5}
	3	67	1.54×10^{-5}
	5	389	1.27×10^{-5}
	8	2978	7.63×10^{-6}
	10	3297	6.48×10^{-6}
	12	5243	6.21×10^{-6}

Table 2. *Cont.*

Scenarios	Number of Components	Number of EM Iterations	MSE
Scenario 4	2	26	1.16×10^{-4}
	3	44	7.54×10^{-5}
	5	266	4.41×10^{-5}
	8	1259	2.67×10^{-5}
	10	1192	2.51×10^{-5}
	12	2055	2.37×10^{-5}

In what follows, we compare the PDF plots of the estimated GMM PDF for three and eight components with the empirical dataset PDF. The impact of changing the sample rates $(1/100, 1/50, 1/20, 1/10, 1/5, 1/2$, and $1)$ and the number of components $(2, 3, 5, 8, 10, 12)$ on the GMM's accuracy MSE and EM computation time is characterized by running the estimation methodology a thousand of times and averaging the results. The average MSE and computation time are plotted for each scenario.

Scenario 1 considers the Core E2E delay data for an NSA 5G testbed in the download stream and for 1024 bytes packet size. This dataset contains 10,000 samples and the PDFs obtained for the GMM for three and eight components are represented in Figure 7. As can be seen in the figure, the adoption of eight components increases the model's accuracy when compared to three.

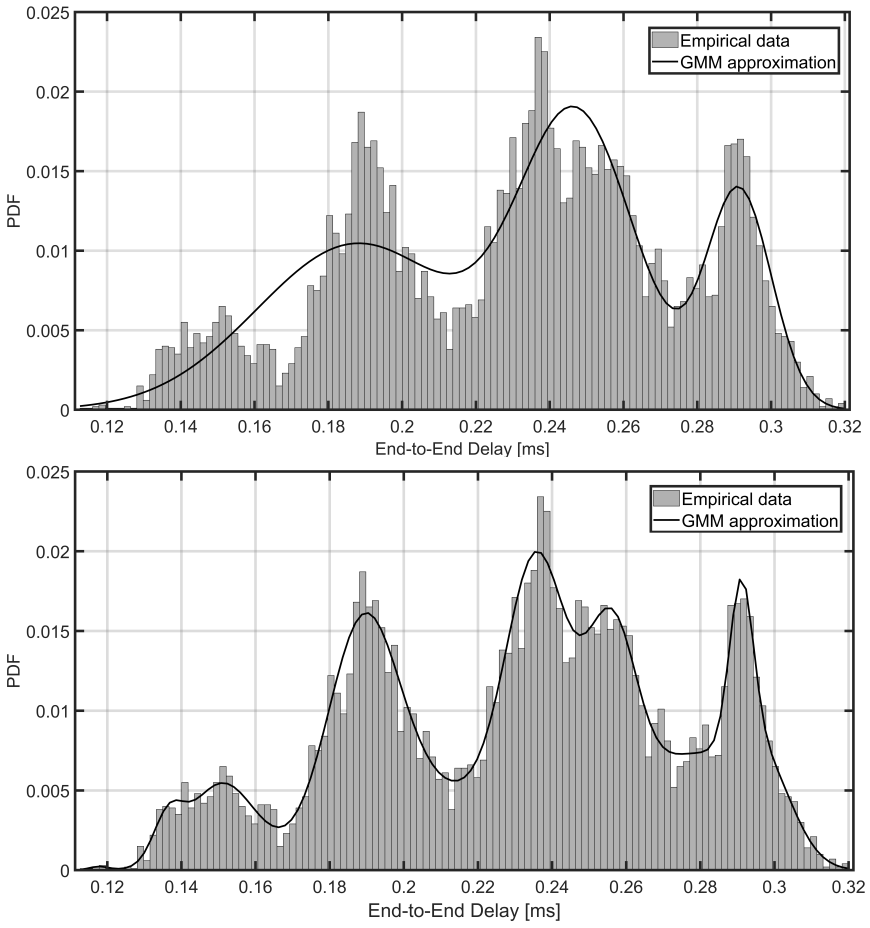

Figure 7. GMM approximation for different number of components. (**a**) 3 GMM components. (**b**) 8 GMM components.

The MSE values for different GMM component numbers and sample rates are summarized in Table 3 to characterize their impact on GMM estimation accuracy. For the same number of GMM components, the MSE decreases as the number of samples increases, which means that a lower error is achieved when a higher number of samples is used. In addition, the MSE value decreases as the number of GMM components increases for a fixed sample rate. This is due to the increase in the number of Gaussian distributions considered in the GMM model.

Table 3. MSE as a function of the number of GMM components and samples.

	$T = 100$	$T = 200$	$T = 500$	$T = 1000$	$T = 2000$	$T = 5000$	$T = 10{,}000$
$K = 2$	2.28×10^{-5}	2.22×10^{-5}	2.09×10^{-5}	1.87×10^{-5}	1.68×10^{-5}	1.43×10^{-5}	1.25×10^{-5}
$K = 3$	1.86×10^{-5}	1.63×10^{-5}	1.44×10^{-5}	1.31×10^{-5}	1.19×10^{-5}	1.09×10^{-5}	8.86×10^{-6}
$K = 5$	1.75×10^{-5}	1.21×10^{-5}	1.07×10^{-5}	8.87×10^{-6}	7.66×10^{-6}	5.60×10^{-6}	5.59×10^{-6}
$k = 8$	1.62×10^{-5}	1.07×10^{-5}	8.79×10^{-6}	7.37×10^{-6}	6.41×10^{-6}	5.55×10^{-6}	4.54×10^{-6}
$k = 10$	1.58×10^{-5}	9.96×10^{-6}	8.33×10^{-6}	6.46×10^{-6}	5.18×10^{-6}	4.34×10^{-6}	3.62×10^{-6}
$k = 12$	1.50×10^{-5}	9.11×10^{-6}	7.59×10^{-6}	5.59×10^{-6}	4.46×10^{-6}	3.44×10^{-6}	2.71×10^{-6}

The computation time is presented in Table 4. As a general trend, the computation time increases with the number of samples for a specific number of GMM components due to the longer sample vector processed by the EM algorithm. For a fixed number of samples, the computation time increases with the number of components due to the increased number of parameters required to compute.

Table 4. GMM computation time [ms] varying the number of GMM components and samples.

	$T = 100$	$T = 200$	$T = 500$	$T = 1000$	$T = 2000$	$T = 5000$	$T = 10{,}000$
$K = 2$	7.30	8.40	14.03	19.17	35.65	86.55	138.81
$K = 3$	8.60	11.34	23.01	38.61	89.32	151.40	263.9
$K = 5$	16.51	35.88	138.13	320.57	739.98	1172.55	1689.99
$k = 8$	53.22	188.04	634.55	1113.33	2990.64	7290.56	17,214.17
$k = 10$	79.05	288.03	1195.34	2224.74	4079.62	13,574.61	38,325.33
$k = 12$	99.84	340.71	1765.19	2554.38	6580.25	25,759.61	60,819.72

Figure 8 illustrates the logarithmic plot of the MSE and computation time per component number for different numbers of E2E delay samples in Scenario 1. Each curve represents a specific number of samples, T. Based on the results, to keep MSE errors below 10^{-5}, it is beneficial to decrease T and increase the number of GMM components, K, as the computational time is more affected by the number of samples than by the number of GMM components.

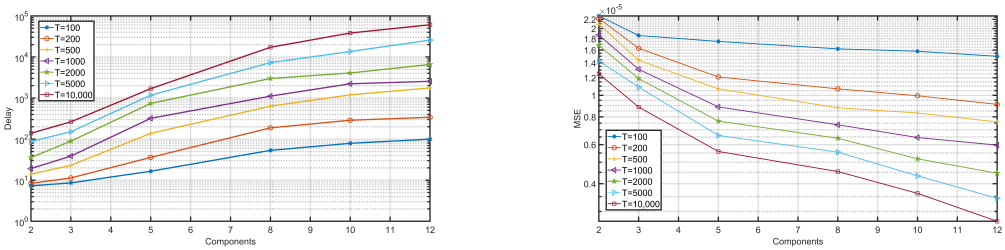

Figure 8. MSE and computation time (delay in milliseconds) for different numbers of GMM components and samples.

Scenario 2 collected Core E2E delay data for a thousand seconds in NSA 5G testbed in the upload stream with 256 bytes of packet size. The dataset contains 10,000 E2E delay samples. The PDFs obtained for three and eight components are represented in Figure 9. Based on the results, the adoption of eight components increases the model's accuracy compared to three.

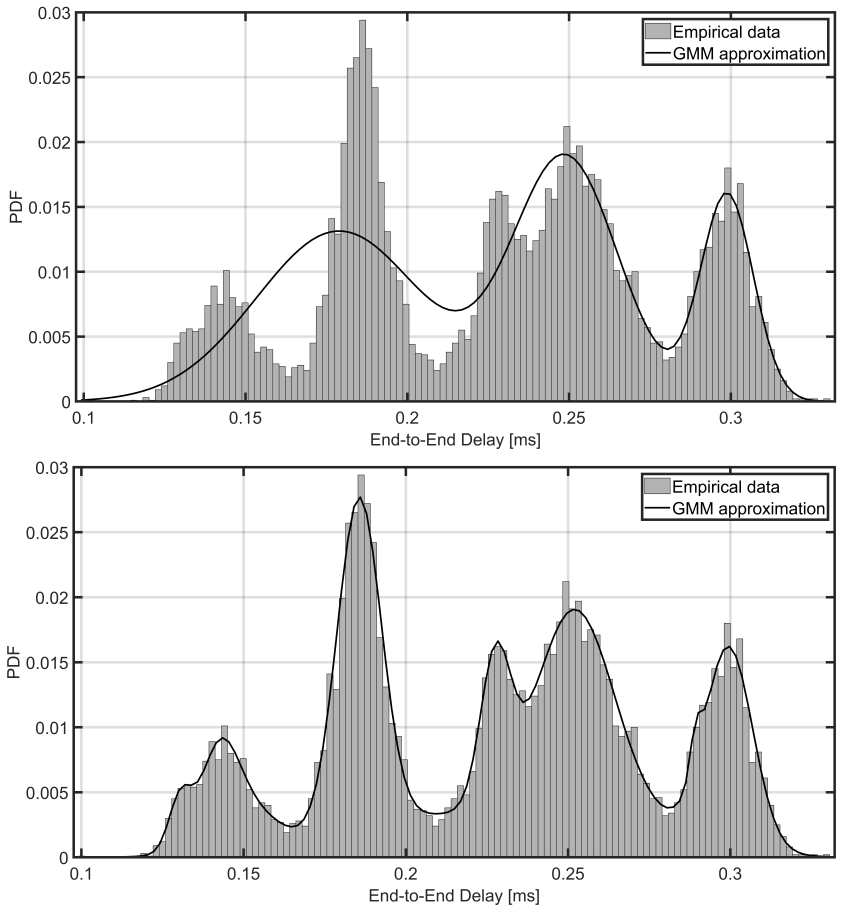

Figure 9. GMM approximation for different number of components. (**a**) 3 GMM components. (**b**) 8 GMM components.

The MSE values for different GMM component numbers and sample rates are summarized in Table 5 to characterize their impact on GMM estimation accuracy. For the same GMM component number, the MSE value decreases as the number of samples increases, which means that lower error occurs when the sample rates increase. In addition, the MSE value decreases with the number of GMM components for a fixed sample rate due to the need of computing more parameters to represent the experimental data accurately.

Table 5. MSE as a function of the number of GMM components and samples.

	$T = 100$	$T = 200$	$T = 500$	$T = 1000$	$T = 2000$	$T = 5000$	$T = 10{,}000$
$K = 2$	3.28×10^{-5}	3.16×10^{-5}	3.05×10^{-5}	3.02×10^{-5}	3.01×10^{-5}	2.98×10^{-5}	2.97×10^{-5}
$K = 3$	2.74×10^{-5}	2.09×10^{-5}	1.82×10^{-5}	1.81×10^{-5}	1.80×10^{-5}	1.77×10^{-5}	0.94×10^{-5}
$K = 5$	2.11×10^{-5}	1.50×10^{-5}	6.30×10^{-6}	5.13×10^{-6}	4.94×10^{-6}	4.83×10^{-6}	4.55×10^{-6}
$k = 8$	1.88×10^{-5}	1.12×10^{-5}	6.24×10^{-6}	4.73×10^{-6}	4.32×10^{-6}	4.19×10^{-6}	2.01×10^{-6}
$k = 10$	1.77×10^{-5}	1.10×10^{-5}	5.90×10^{-6}	2.81×10^{-6}	1.87×10^{-6}	1.33×10^{-6}	1.11×10^{-6}
$k = 12$	1.74×10^{-5}	1.06×10^{-5}	5.50×10^{-6}	2.72×10^{-6}	1.61×10^{-6}	9.96×10^{-6}	8.23×10^{-6}

The computation time represented for Scenario 2 is represented in Table 6. The computation time increases with the number of samples for a specific number of components.

Table 6. GMM computation time [ms] varying the number of GMM components and samples.

	$T = 100$	$T = 200$	$T = 500$	$T = 1000$	$T = 2000$	$T = 5000$	$T = 10{,}000$
$K = 2$	3.40	3.98	7.02	8.89	14.62	30.28	53.07
$K = 3$	4.16	5.95	13.50	25.70	36.63	74.17	99.52
$K = 5$	27.21	44.13	115.34	330.02	520.93	1329.19	1992.44
$k = 8$	38.78	72.12	297.23	641.12	944.55	3603.70	5117.84
$k = 10$	46.36	173.71	614.38	1808.22	3627.43	1162.09	21,256.1
$k = 12$	67.97	254.89	1073.23	2630.99	6973.03	21,931.42	43,523.65

Figure 10 illustrates the logarithmic plot of the MSE and computation time per component number for different numbers of samples in Scenario 2. Each curve represents a specific number of samples. Based on the results, to keep MSE errors below 10^{-5}, it is beneficial to decrease the sample numbers (T) and increase the GMM components number (K), as the computational time is more affected by the number of samples than by the number of GMM components.

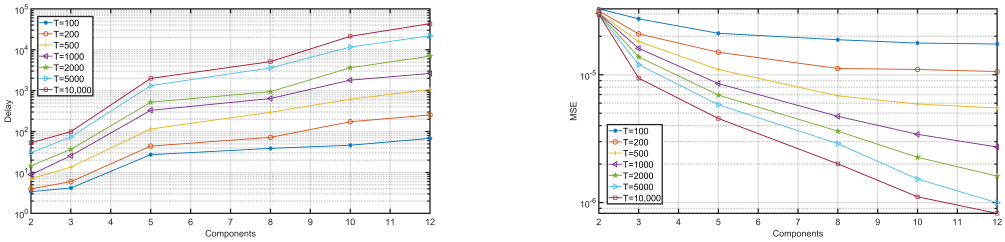

Figure 10. MSE and computation time (delay in milliseconds) for different number of GMM components and samples.

In Scenario 3, we consider RAN E2E delay samples in an SA 5G testbed and for a download stream obtained with 1024 bytes packet size. The dataset collects 100,000 data samples and the PDFs obtained for three and eight GMM components are represented in Figure 11.

The MSE values for different GMM component numbers and sample rates are summarized in Table 7. For the same number of GMM components, the MSE value decreases as the number of samples increases. In addition, the MSE value decreases with the number of GMM components, as previously observed.

Table 7. MSE as a function of the number of GMM components and samples.

	$T = 1000$	$T = 2000$	$T = 5000$	$T = 10{,}000$	$T = 20{,}000$	$T = 50{,}000$	$T = 100{,}000$
$K = 2$	1.84×10^{-5}	1.79×10^{-5}	1.75×10^{-5}	1.73×10^{-5}	1.69×10^{-5}	1.66×10^{-5}	1.62×10^{-5}
$K = 3$	1.63×10^{-5}	1.59×10^{-5}	1.53×10^{-5}	1.46×10^{-5}	1.37×10^{-5}	1.29×10^{-5}	1.24×10^{-5}
$K = 5$	1.33×10^{-5}	1.28×10^{-5}	1.21×10^{-5}	1.16×10^{-5}	1.11×10^{-5}	1.06×10^{-5}	1.01×10^{-5}
$k = 8$	10.52×10^{-6}	10.02×10^{-6}	9.48×10^{-6}	9.15×10^{-6}	8.52×10^{-6}	8.08×10^{-6}	7.55×10^{-6}
$k = 10$	9.16×10^{-6}	8.67×10^{-6}	8.22×10^{-6}	7.88×10^{-6}	7.37×10^{-6}	6.96×10^{-6}	6.58×10^{-6}
$k = 12$	7.85×10^{-6}	7.41×10^{-6}	7.14×10^{-6}	6.84×10^{-6}	6.58×10^{-6}	6.24×10^{-6}	6.03×10^{-6}

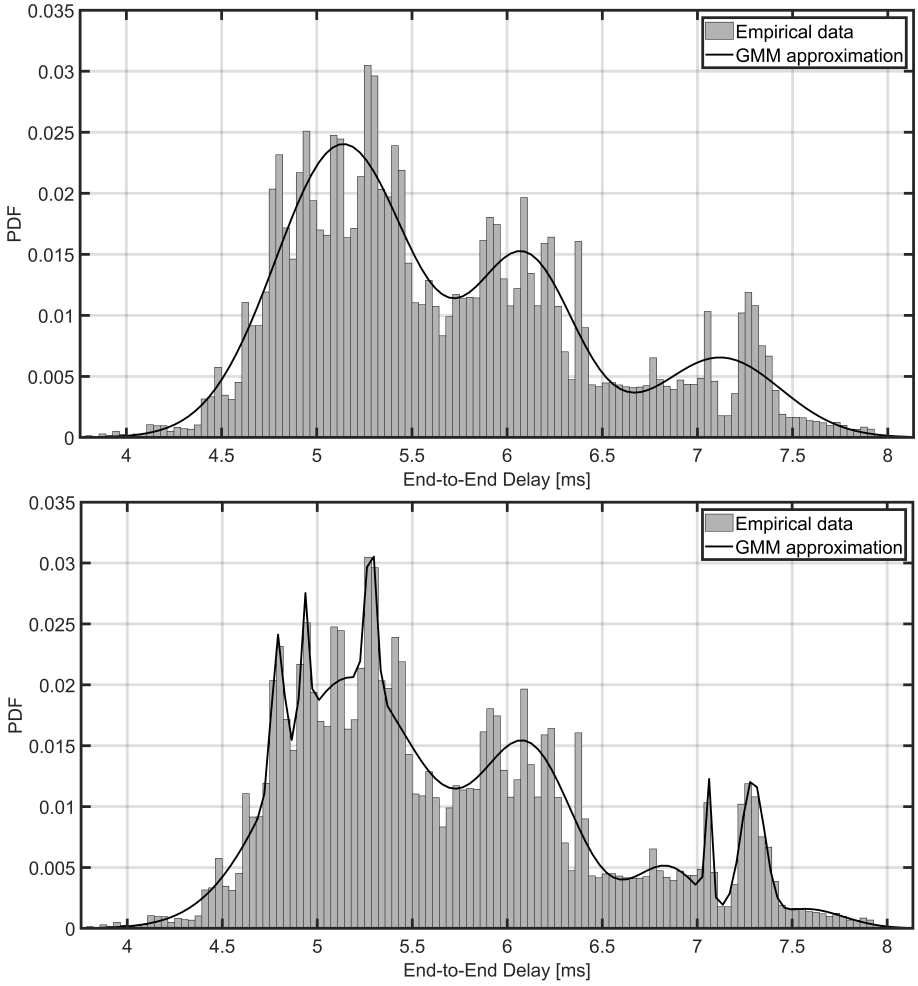

Figure 11. GMM approximation for different number of components. (**a**) 3 GMM components. (**b**) 8 GMM components.

The computation time is presented in Table 8.

Table 8. GMM computation time [ms] varying the number of GMM components and samples.

	$T = 1000$	$T = 2000$	$T = 5000$	$T = 10{,}000$	$T = 20{,}000$	$T = 50{,}000$	$T = 100{,}000$
$K = 2$	11.30	19.74	41.43	74.97	115.30	222.79	507.28
$K = 3$	37.26	107.95	231.27	417.43	688.12	1100.44	1997.72
$K = 5$	255.25	456.99	847.52	1383.48	2369.19	5254.35	12,277.65
$k = 8$	1900.31	3160.08	4914.20	10,821.52	22,088.32	46,589.28	73,665.23
$k = 10$	4155.29	7202.56	9949.18	21,533.01	36,724.32	83,311.76	161,840.42
$k = 12$	5152.08	10,340.62	20,460.15	34,144.78	87,300.42	154,495.97	201,126.33

Figure 12 illustrates the logarithmic plot of the MSE and computation time per component number for different sample numbers (sample rates) in Scenario 3.

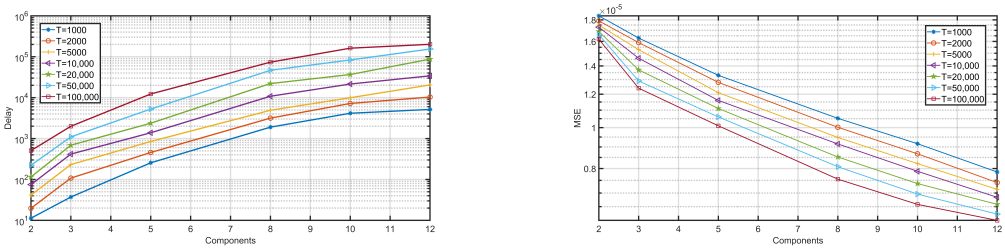

Figure 12. MSE and computation time (delay in milliseconds) for different number of GMM components and samples.

In Scenario 4, we consider the RAN E2E delay data in an SA 5G testbed for the upload stream and considering a packet size of 128 bytes. The PDFs obtained for the three and eight GMM components are represented in Figure 13. As can be observed, the PDF of the E2E delay is no similar to the ones obtained for Scenarios 1–3.

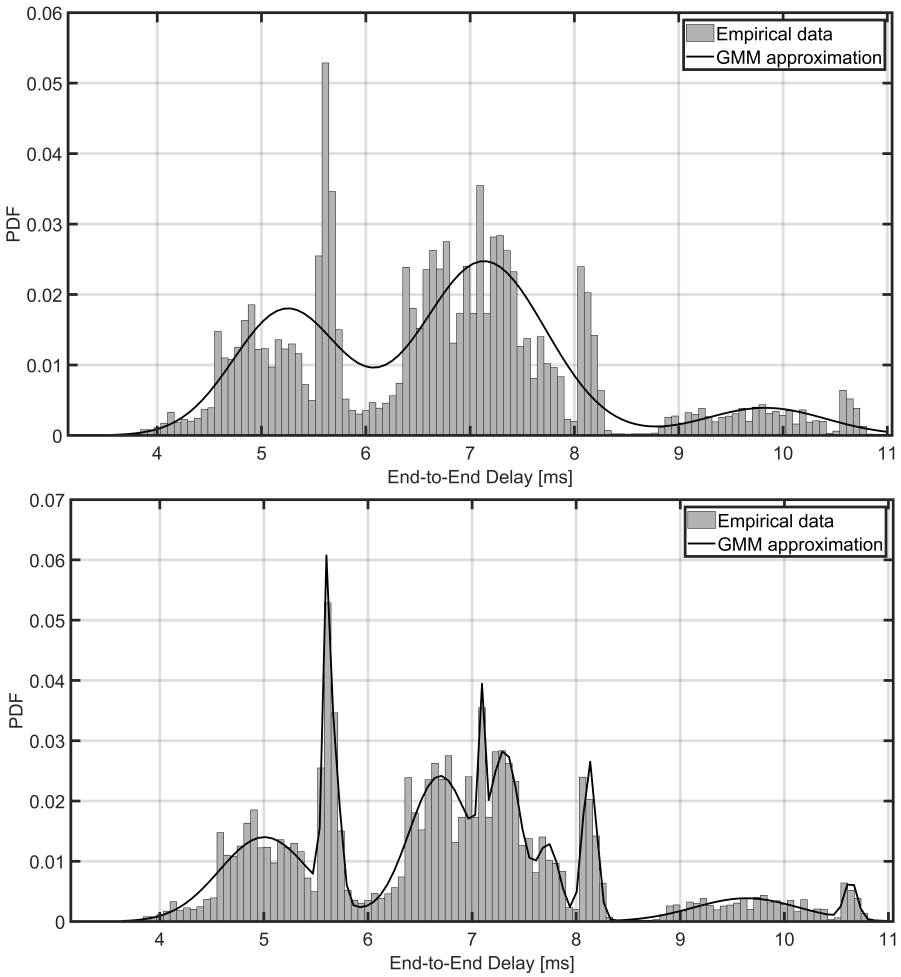

Figure 13. GMM approximation for different number of components. (**a**) 3 GMM components. (**b**) 8 GMM components.

The MSE values for the different numbers of GMM components are summarized in Table 9.

Table 9. MSE as a function of the number of GMM components and samples.

	$T = 1000$	$T = 2000$	$T = 5000$	$T = 10{,}000$	$T = 20{,}000$	$T = 50{,}000$	$T = 100{,}000$
$K = 2$	1.12×10^{-4}	1.11×10^{-4}	1.09×10^{-4}	1.07×10^{-4}	1.04×10^{-4}	1.03×10^{-4}	1.00×10^{-4}
$K = 3$	8.89×10^{-5}	8.72×10^{-5}	8.57×10^{-5}	8.29×10^{-5}	8.02×10^{-5}	7.70×10^{-5}	7.37×10^{-5}
$K = 5$	6.41×10^{-5}	6.14×10^{-5}	5.91×10^{-5}	5.60×10^{-5}	5.16×10^{-5}	4.71×10^{-5}	4.37×10^{-5}
$k = 8$	3.94×10^{-5}	3.65×10^{-5}	3.42×10^{-5}	3.24×10^{-5}	3.04×10^{-5}	2.85×10^{-5}	2.71×10^{-5}
$k = 10$	3.47×10^{-5}	2.25×10^{-5}	3.01×10^{-5}	2.82×10^{-5}	2.63×10^{-5}	2.46×10^{-5}	2.36×10^{-5}
$k = 12$	3.00×10^{-5}	2.79×10^{-5}	2.62×10^{-6}	2.48×10^{-5}	2.32×10^{-5}	2.21×10^{-5}	2.05×10^{-5}

The computation times for Scenario 4 is presented in Table 10. Once again, we observe the same trends as the ones obtained for Scenarios 1–3.

Table 10. GMM computation time [ms] varying the number of GMM components and samples.

	$T = 1000$	$T = 2000$	$T = 5000$	$T = 10{,}000$	$T = 20{,}000$	$T = 50{,}000$	$T = 100{,}000$
$K = 2$	7.16	11.74	23.08	39.26	58.30	119.10	187.09
$K = 3$	19.38	34.16	64.28	112.24	198.35	351.52	505.20
$K = 5$	93.09	167.41	288.08	410.98	722.98	1643.57	2250.46
$k = 8$	444.28	716.64	1332.91	2109.27	4136.47	9490.86	12,865.59
$k = 10$	811.68	1307.54	2820.18	4903.51	7762.40	22,874.94	32,506.24
$k = 12$	1215.34	2248.68	4520.36	9075.60	15,267.56	48,855.78	61,951.31

Figure 14 illustrates the logarithmic plot of the MSE and computation time per component number for different sample rates in Scenario 4. As can be seen, the performance of the estimation methodology effectively depends on the 5G topology, but also on the number of GMM components and the number of input samples.

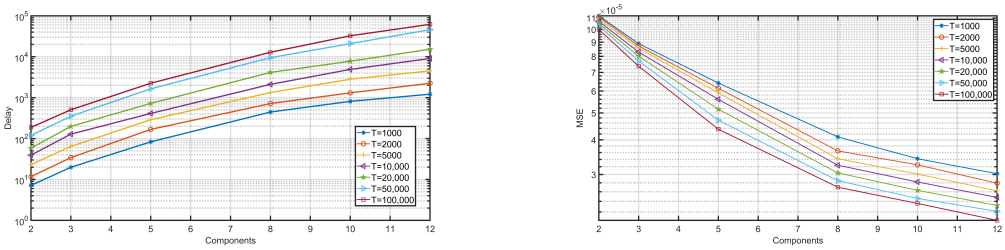

Figure 14. MSE and computation time (delay in milliseconds) for different number of GMM components and samples.

As confirmed by the results presented in this section, the GMM can be effectively used to estimate the E2E delay in a short amount of time, validating the initial hypothesis. The proposed methodology focuses on a more detailed estimation of the E2E delay, where instead of estimating a bound as in NC methods, or instead of computing an expected value of the delay as in QT models, we estimate the distribution of the E2E delay. However, the parameters adopted in the GMM model strongly influence the accuracy and computation time of the estimation. As a final remark of the results described in this section, we conclude that:

- By increasing the number of GMM components the number of parameters to estimate also increase, so the computing time. The computation time increases approximately exponentially with the number of components, although the estimation accuracy increases in a smaller scale;

- By increasing the number of components in all scenarios, the average number of EM iterations for reaching the convergence threshold also increases. Due to the sensitivity of this parameter with regards to the difference of estimates, EM takes more iterations in the scenarios with larger deviations in the E2E delay samples;
- The number of samples causes a tremendous impact on the estimation computation time. Although the accuracy of the estimation is significantly reduced for a smaller amount of samples, the computation time can be significantly reduced as the number of GMM components increases;
- Although a linear relation between the MSE and the number of GMM components is not found, they always exhibit an inverse trend;
- Although there is no linear relation between computation time and the number of GMM components, they always exhibit a direct trend.

6. Conclusions

This paper proposes a method to model the E2E delay of 5G networks with a GMM model. As a numerical and iterative method, the EM algorithm is employed to estimate the mixture model parameters. The estimation methodology is evaluated based on the number of GMM components, the number of samples, and the computation time.

The results show that higher accuracy is achieved as the number of samples and GMM components increases. However, the computation time also increases approximately exponentially with the number of samples and components, as identified by the trade-off between the model's mean square error and its computational time presented in the performance evaluation section. Future work includes the adoption of machine-learning approaches to identify the GMM model parameters through unsupervised deep-learning neural networks.

Author Contributions: Conceptualization, D.F. and R.O.; methodology, D.F. and R.O.; validation, D.F.; formal analysis, D.F. and R.O.; investigation, D.F. and R.O.; writing—review and editing, D.F. and R.O.; visualization, D.F.; supervision, R.O.; project administration, R.O.; funding acquisition, R.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Fundação para a Ciência e Tecnologia (FCT) under the projects 2022.08786.PTDC and UIDB/50008/2020.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hung, M. Leading the IoT Gartner Insights on How to Lead in a Connected World. *Gart. Res.* **2017**, *1*, 1–29.
2. Analyst, A.O.P. Nokia Ava Network Data Analytics Function. Available online: https://www.nokia.com/networks/bss-oss/nwdaf/index.php (accessed on 28 November 2022).
3. Pandey, K.; Arya, R. Robust Distributed Power Control with Resource Allocation in D2D Communication Network for 5G-IoT Communication System. *Int. J. Comput. Netw. Inf. Secur.* **2022**, *14*, 73–81. [CrossRef]
4. Afolabi, I.; Taleb, T.; Samdanis, K.; Ksentini, A.; Flinck, H. Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2429–2453. [CrossRef]
5. Banavalikar, B.G. Quality of Service (QoS) for Multi-Tenant-Aware Overlay Virtual Networks. US Patent 10,177,936, 28 March 2014.
6. Yao, J.; Pang, S.; Rodrigues, J.J.P.C.; Lv, Z.; Wang, S. Performance Evaluation of MPTCP Incast Based on Queuing Network. *IEEE Trans. Green Commun. Netw.* **2022**, *6*, 695–703. [CrossRef]
7. Ye, Y.; Lee, B.; Flynn, R.; Xu, J.; Fang, G.; Qiao, Y. Delay-Based Network Utility Maximization Modelling for Congestion Control in Named Data Networking. *IEEE/ACM Trans. Netw.* **2021**, *29*, 2184–2197. [CrossRef]
8. Diez, L.; Alba, A.M.; Kellerer, W.; Agüero, R. Flexible Functional Split and Fronthaul Delay: A Queuing-Based Model. *IEEE Access* **2021**, *9*, 151049–151066. [CrossRef]

9. Adamuz-Hinojosa, O.; Sciancalepore, V.; Ameigeiras, P.; Lopez-Soler, J.M.; Costa-Pérez, X. A Stochastic Network Calculus (SNC)-Based Model for Planning B5G URLLC RAN Slices. *IEEE Trans. Wirel. Commun.* **2022**. . [CrossRef]
10. Boroujeny, M.K.; Mark, B.L. Design of a Stochastic Traffic Regulator for End-to-End Network Delay Guarantees. *IEEE/ACM Trans. Netw.* **2022**, 1–13 . [CrossRef]
11. Mei, M.; Yao, M.; Yang, Q.; Qin, M.; Jing, Z.; Kwak, K.S.; Rao, R.R. On the Statistical Delay Performance of Large-Scale IoT Networks. *IEEE Trans. Veh. Technol.* **2022**, *71*, 8967–8979. [CrossRef]
12. Chinchilla-Romero, L.; Prados-Garzon, J.; Ameigeiras, P.; Muñoz, P.; Lopez-Soler, J.M. 5G Infrastructure Network Slicing: E2E Mean Delay Model and Effectiveness Assessment to Reduce Downtimes in Industry 4.0. *Sensors* **2021**, *22*, 229. [CrossRef]
13. Ye, Q.; Zhuang, W.; Li, X.; Rao, J. End-to-End Delay Modeling for Embedded VNF Chains in 5G Core Networks. *IEEE Internet Things J.* **2019**, *6*, 692–704. [CrossRef]
14. McLachlan, G.J.; Lee, S.X.; Rathnayake, S.I. Finite Mixture Models. *Annu. Rev. Stat. Its Appl.* **2019**, *6*, 355–378. [CrossRef]
15. Reynolds, D.A. Gaussian Mixture Models. *Encycl. Biom.* **2009**, *741*, 659–663.
16. Moon, T.K. The Expectation-Maximization Algorithm. *IEEE Signal Process. Mag.* **1996**, *13*, 47–60. [CrossRef]
17. Fadhil, D.; Oliveira, R. A Novel Packet End-to-End Delay Estimation Method for Heterogeneous Networks. *IEEE Access* **2022**, *10*, 71387–71397. [CrossRef]
18. Yang, M.S.; Lai, C.Y.; Lin, C.Y. A Robust EM Clustering Algorithm for Gaussian Mixture Models. *Pattern Recognit.* **2012**, *45*, 3950–3961. [CrossRef]
19. Lawrence, E.; Michailidis, G.; Nair, V. Maximum Likelihood Estimation of Internal Network Link Delay Distributions Using Multicast Measurements. In Proceedings of the 37th Conference on Information Sciences and Systems , Baltimore, MD, USA, 12–14 March 2003.
20. Orellana, R.; Carvajal, R.; Agüero, J.C. Maximum Likelihood Infinite Mixture Distribution Estimation Utilizing Finite Gaussian Mixtures. *IFAC-PapersOnLine* **2018**, *51*, 706–711. [CrossRef]
21. Barazandeh, B.; Razaviyayn, M. On the Behavior of the Expectation-Maximization Algorithm for Mixture Models. In Proceedings of the 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Anaheim, CA, USA, 26–29 November 2018; pp. 61–65.
22. Huang, T.; Peng, H.; Zhang, K. Model Selection for Gaussian Mixture Models. *Stat. Sin.* **2017**, 147–169. [CrossRef]
23. Mistry, D.; Joshi, S.; Agrawal, N. A Novel Jitter Separation Method Based on Gaussian Mixture Model. In Proceedings of the 2015 International Conference on Pervasive Computing (ICPC), Pune, India, 8–10 January 2015; pp. 1–4.
24. Plasse, J.H. The EM Algorithm in Multivariate Gaussian Mixture Models Using Anderson Acceleration. Ph.D. Thesis, Worcester Polytechnic Institute, Worcester, MA, USA, 2013.
25. O'Hagan, A.; Murphy, T.B.; Gormley, I.C. Computational Aspects of Fitting Mixture Models Via the Expectation-Maximization Algorithm. *Comput. Stat. Data Anal.* **2012**, *56*, 3843–3864. [CrossRef]
26. Rahman, M.L.; Zhang, J.A.; Huang, X.; Guo, Y.J.; Lu, Z. Gaussian-Mixture-Model Based Clutter Suppression in Perceptive Mobile Networks. *IEEE Commun. Lett.* **2021**, *25*, 152–156. [CrossRef]
27. Cerroni, W.; Foschini, L.; Grabarnik, G.Y.; Shwartz, L.; Tortonesi, M. Estimating Delay Times Between Cloud Datacenters: A Pragmatic Modeling Approach. *IEEE Commun. Lett.* **2018**, *22*, 526–529. [CrossRef]
28. Rischke, J.; Sossalla, P.; Itting, S.; Fitzek, F.H.P.; Reisslein, M. 5G Campus Networks: A First Measurement Study. *IEEE Access* **2021**, *9*, 121786–121803. [CrossRef]
29. Fidge, C.J. Timestamps in Message-Passing Systems that Preserve the Partial Ordering. In Proceedings of the 11th Australian Computer Science Conference, Perth, Australia, 3–5 February 1988.
30. Manivannan, D.; Singhal, M. Asynchronous Recovery Without Using Vector Timestamps. *J. Parallel Distrib. Comput.* **2002**, *62*, 1695–1728. [CrossRef]