End-to-End Latency Analysis in Wireless Networks with Queuing Models for General Prioritized Traffic

Philipp Schulz[*], Lyndon Ong[i], Paul Littlewood[i], Bashar Abdullah[i], Meryem Simsek[i], and Gerhard Fettweis[*]
[*]Barkhausen Institut, Dresden, Germany. Email: {firstname.lastname}@barkhauseninstitut.org.
[i]International Computer Science Institute, Berkeley, California, USA. Email: {pschulz, simsek}@icsi.berkeley.edu. [i]Ciena, 7035 Ridge Road Hanover, MD 21076 E-mail: {lyong, plittlew, babdulla}@ciena.com.

Abstract—Future generations of mobile networks are expected to serve a multitude of applications with different requirements and traffic characterizations. Through network slices, the applications shall be even served on the same physical infrastructure. However, this requires a sophisticated network management, for an efficient sharing of the common resources and for meeting the demanding requirements of applications with very stringent requirements, e.g., ultra-high reliability, low-latency. In this regard, mathematical models can help in understanding and optimizing the network performance. By introducing queuing models for general systems and scheduling policies for heterogeneous traffic, this paper is an important step for accurate modeling of general systems and towards slice configuration and optimization.

Index Terms—Queuing Models, Low-Latency, URLLC, 5G, Access Networks

I. Introduction

With the fifth generation (5G), mobile networks are confronted with heterogeneous applications, characterized by different requirements as well as revised traffic patterns. For instance, mobile broadband applications (e.g., entertainment) will evolve further, demanding more throughput and capacity from the network. On the other hand, there are mission-critical use cases (e.g., in factory automation, robotics, road traffic) requiring ultra-reliable low latency communications (URLLC) [1]. Moreover, these different applications may be served on the same physical infrastructure sharing common resources, which motivates the concept of network slices, i.e., differently configurable logical networks for the different applications running on a common infrastructure. Technologies like software defined networks (SDN), network function virtualization (NFV), and mobile edge computing (MEC) are enablers for a flexible network architecture that can be tailored to the specific needs of the different applications. However, the management and configuration of these slices is not a trivial task.

In particular, URLLC applications require that packets are delivered correctly within a certain end-to-end (E2E) latency bound. This has to be guaranteed with a very high probability to ensure that the application works correctly. Otherwise, systems may shut down and cause, e.g., financial damage, because production is interrupted. Moreover, it becomes cumbersome or even infeasible due to the complexity of the systems

This research was co-financed with tax money on the foundation of the budget decided upon by the delegates of the state parliament of Saxony/Germany. This material is also based upon work supported by the National Science Foundation under Grant No. 1745410.

978-1-7281-2373-8/19/$31.00 ©2019 IEEE

and the ultra-low probabilities, which require a huge number of samples to be validated, and hence, become infeasible.

Therefore, recent work, e.g., [2], [3], focused on latency modeling with respect to URLLC. The authors in [2] investigate wireless fading channels with finite block length channel coding. In [3], the need for latency modelling for URLLC traffic is stated and queuing delay violation is identified as an important issue for reliability.

In this regard we propose a general mathematical framework, based on queuing networks, to assess the E2E latency (i.e., sojourn time in queuing terminology) of any given network topology. It can be used to evaluate and understand the performance of any network and also for network optimization. This article focuses on how to deal with general nodes in the network and adds the following contributions:

1) A numerical method to determine the waiting time distribution (and thus, the sojourn time) in $GI\{GI\{1$ queues is provided, such that traffic scenarios, e.g., URLLC traffic, with arbitrary independent inter-arrival and service time distributions can be studied. To the best of our knowledge, this method has not been used before.
2) Furthermore, heterogeneous traffic, i.e., traffic from multiple applications with different requirements, priorities, and different characteristics, is considered.
3) In this regard, different scheduling concepts, namely dedicated resources, priority queuing with and without preemption as well as weighted fair queuing, are modeled and compared with each other to demonstrate that common scheduling methods can be applied to our proposed models and to compare their latency behavior.
4) Finally, we verify our models by extensive simulations.

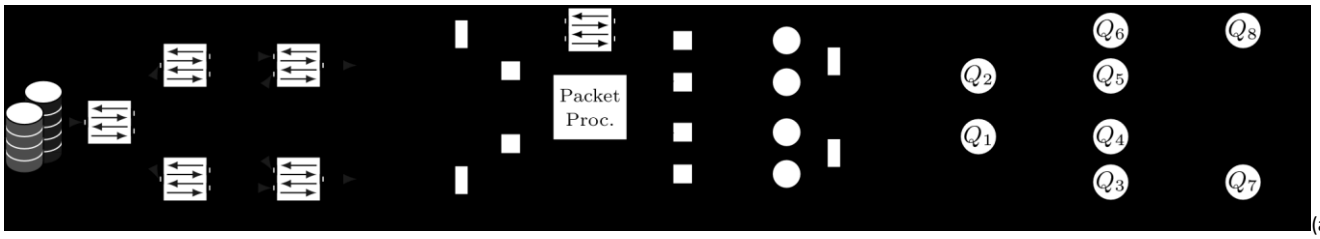
II. System Model

A. Notation and Definitions

Throughout the article, the following notations will be used. Random variables (RV) will be denoted by capital letters X and their realizations as lowercase letters x. The corresponding probability density function (pdf) and cumulative distribution function (cdf) are depicted as f_X and F_X, respectively. The operator $\circ$ will be used for the convolution of two functions, i.e.,

$$(f_X \circ f_Y)(t) = \int_{-\infty}^{\infty} f_X(\tau) f_Y(t-\tau)\, d\tau. \quad (1)$$

Furthermore, $\mathbf{1}_A$ denotes the indicator function for any set A.



(a)

(b)

Fig. 1. Illustration of the scenario and the modeling: (a) the network topology including data centers, network nodes, and RAN, (b) the model of the ESsEthernet switches, and (c) the resulting queuing network of the ESs.

B. Scenario

In this work, the E2E latency in a mobile network including the radio access network (RAN) is of interest. The research is motivated by the scenario described in this section.

1) Applications: 5G is envisioned to support or even to enable a multitude of applications with diverging and ambitious requirements. On the one hand, there is the further enhancement of mobile broadband (eMBB) applications, such as 3D or UHD video streaming, which are mainly driven by the need for high data rates and capacity. On the other hand, there are massive machine type communications (mMTC) and URLLC applications. Both are characterized by the fact that mainly machines rather than human beings are communicating with each other. However, in mMTC a huge number of devices, e.g., sensors, are connected, asking for energy-efficient communications and wide coverage. In contrast, URLLC use cases comprise critical communications, where high reliability (error rates down to $10^{\acute{}9}$) and low latency (1ms)down to error rates of $10^{\acute{}9}$ and low latency of 1ms are crucial [1]. Applications in the field of factory automation and intelligent transport system are among the most promising examples.

To this end, a flexible architecture is required, which does not address only one of these applications, but can be tailored to their specific needs. Therefore, it is envisioned to establish differently configured network slices, which are logical networks that run on the same infrastructure, such that the above-mentioned applications can be served simultaneously by the same mobile transport network.

2) Network Topology and Deployment: The aim of the modeling is to investigate mobile network scenarios, such as the one sketched in Fig. 1(a). Here, the illustrated network consists of base stations (BS), Ethernet Switches (ES), and a cloud server (CS). All these elements are characterized by the fact, that they are systems with shared resources, which can cause delay. This is why queuing models appear to be suitable. However, each element may be characterized by different queuing properties, such as arrival and services processes, capacities or scheduling policies. In this scenario, the E2E latency on a path through the network is of interest for any of these properties. For this purpose, we provide an analytical model with a particular focus on the scheduling behavior of the nodes in this work.

3) Ethernet Switch Behavior: As an example, it is sketched how the ESs handle the traffic that is routed through the element. The structure of such an ES is illustrated in Fig. 1(b). At each of its inputs, there are meters, that drop the traffic, which violates service level agreements (SLA). Afterwards, packets go through the packet processor, which determines to which output queue a packet is to be forwarded. The processing is designed such that it can process each packet in constant time. Each output of the ES has multiple queues, which collect the packets of different classes or priorities. Before a packet reaches the queues, weighted random early discard (WRED, [4], [5]) is applied to avoid congestion by randomly dropping packets based on the instantaneous queue utilization, configured weights, thresholds and drop probabilities.

This work focuses on the queuing part and the different possibilities to schedule the queues at each output. In the final framework, a real world scenario, such as the one in Fig. 1(a) may be mapped to a queuing network. Here, each ES in 1(a) was replaced by the model in Fig. 1(b) to obtain the queuing network of the ESs in Fig. 1(c). The proposed model will enable network managers and operators to plan and optimize their networks.

C. Traffic Modeling

Since queuing systems are sensitive to the assumed traffic characteristics, i.e., in queuing vocabulary the involved arrival and service process, it is of high importance to choose appropriate models. Since MTC is a relatively new field, research on the traffic modeling is still ongoing. However, there are already models considered by the third Generation Partnership Project (3GPP), e.g., the models in [6] and [7], which are suitable for mMTC, where a huge number of sensors send data synchronously or asynchronously. 3GPP also proposes modeling assumptions for URLLC traffic [8]. There $M\{D\{1$ is considered for critical applications.

That is the reason why $M\{D\{1$ is also assumed in this work. However, by proposing an algorithm for $GI\{GI\{1$ queues, the model is kept general enough to apply it to $M\{D\{1$ as well as to more sophisticated traffic models, when they come up.

D. Queuing Network

The overall aim of this work is to analyze the E2E delay with the help of queuing networks, which is briefly sketched in this subsection.

All network elements in the scenario are mapped to one or multiple queues, leading to a queuing network consisting of a fixed number M P N of queues $Q_1,\ldots,Q_M$. Let M " t1,…,Mu denote the index set of these queues. Traffic arrives at the overall queuing network with a mean arrival rate α according to a

inter-arrival time distribution A, which is specified later. Arriving objects enter the network at queue Q_i with probability p_{0i}, i P M. At each queue Q_i, an object experiences a service time S_i, with a distribution B with mean [redacted], which is specified later. Once an object is processed in queue Q_i, it will be forwarded to queue Q_j with probability p_{ij} or leaves the network with probability p_{i0}, i,j P M. Fig. 1(c) illustrates the queuing network model for the scenario under investigation.

At each queue Q_i, an object may experience a waiting time W_i, due to other objects being served before, in addition to its service time S_i. Furthermore, any additional delays D_i, may be introduced by a realistic hardware implementation. The overall time, an object spends at a single node, denoted as *sojourn time* J_i, is the sum of these components

$$J_i \text{“ } W_i \text{` } S_i \text{` } D_i, \qquad i \text{ P N.} \tag{2}$$

Now, let q :“ pq_1,…,q_kq P M^k denote the *path* Q_{q1}, Q_{q2}, ..., Q_{qk} in the network. To derive E2E latency, the overall sojourn time J_q along a path is of interest.

$$J_q \text{“ } [\text{redacted}] \text{p} W_{qi} \text{` } S_{qi} \text{` } D_{qi} \text{q “ } W_q \text{`} S_q \text{`} D_q \tag{3}$$

Whereas the additional delays D_i are initially assumed to be independent, the waiting times W_i in a queuing network are not independent in general. However, according to *Kleinrock's independency approximation* [9], dependency is negligible if the network is dense enough. Exploiting this approximation, the pdf of the overall waiting time W_q along the path q, can be obtained by convolution of the single pdfs.

$$f_{W_q} \text{“ } f_{W_{q1}} \text{˚ ····· ˚ } f_{W_{qk}} \text{“: ˚ } \overset{k}{\underset{i\text{“}1}{f_{W_{qi}}}} \tag{4}$$

For the service time a full dependence is assumed, i.e., the service time S_i of an object at queue Q_i is assumed to be a scaled version of an auxiliary initially drawn service time S_0. This way, a large object will experience a long service at each visited node, possibly scaled by the processing speed of that node. Thus,

$$S_q \text{“ ~} [\text{redacted}]\text{, ,} \tag{5}$$

which translates to

$$f_{S_q}\text{ptq “ } \mu_q f_{S_0}\text{p}\mu_q t\text{q,} \tag{6}$$

for its pdf. Putting everything together, this leads to the pdf of the overall sojourn time along path q

$$f_{J_q}\text{ptq “ } \overset{k}{\underset{i\text{“}1}{\text{˚ } f_{W_{qi}}}}\text{ptq ˚ } \mu_q f_{S_0}\text{p}\mu_q t\text{q ˚ } f_{D_q}\text{ptq.} \tag{7}$$

The remaining question, how to obtain the pdfs $f_{W_{qi}}$ for general inter-arrival and service time distributions and for different scheduling policies, is the focus of this work and will be answered in the subsequent sections.

Algorithm 1: Waiting time distribution for $GI\{GI\{1$.

input : f_S,f_T,tol output: Waiting Time pdf f_W

initialize: [redacted] q Ð 8, $\Delta^{p\acute{}1q}$ Ð 8 ; f_U Ð Convolve(f_S,f_T) ; while $\Delta^{p i q}$ ą tol and $\Delta^{p i q}$ ď $\Delta^{p i\text{-}1q}$ do

  i Ð i ˜ 1;

  $f_W^{p i q}$ Ð Convolve($f_W^{p i\acute{}1q}$,f_U);

  $f_W^{p i q}$ Ð MoveNegativePartToZero($f_W^{p i q}$);

  $\Delta^{p i q}$ Ð Norm($f_W^{p i q}$ ´ $f_W^{p i\acute{}1q}$);

end

E. Waiting Time Distribution in $GI\{GI\{1$ Queues

Let the RVs S_n, T_n, and W_n denote the service time, the inter-arrival time, and the waiting time of the nth object in the considered first-in first-out (FIFO) queuing system, respectively. The service times and the inter-arrival times are both assumed to be i.i.d. RVs, i.e., all S_n and T_n are distributed according to the same given pdfs f_S and f_T, respectively. Based on that, the aim is to derive the steady state distribution of the waiting time, i.e.,

$$[\text{redacted}] \tag{8}$$

It is a common approach [10] to define the auxiliary RV U_n by

$$U_n \text{:“ } S_n \text{´ } T_{n\text{`}1}, \tag{9}$$

which inherits the i.i.d. property. Thus, the pdf of U_n can be obtained by the following convolution

$$f_U\text{ptq “ p}f_S \text{˚ } f_T\text{qptq} \overset{\dot{z}}{\underset{\acute{}8}{\text{`}8}} \text{“}f_S\text{p}\tau\text{q}f_T\text{p}\tau \text{´ } t\text{qd}\tau. \tag{10}$$

The auxiliary RV U_n helps to relate the waiting time of an object, to the waiting time of its predecessor,

$$W_{n\text{`}1} \text{“ max}\text{t}W_n \text{` } U_n\text{,0u,} \tag{11}$$

which is known as *Lindley's equation*. This can also be written as the functional equation

$$f_{W_{n\text{`}1}} \text{“ } \varphi\text{p}f_{W_n} \text{˚ } f_{U_n}\text{q,} \tag{12}$$

where the operator φ pulls the negative part of a function f to a dirac impulse δ_0 at zero, i.e.,

$$\varphi f \text{:“ } \mathbf{1}_{\text{r}0,8\text{q}} \text{¨ } f \text{` ~} \dot{z}_{\text{p}\acute{}8,0\text{q}} f\text{ptqd}t \text{¸ ¨ } \delta_0. \tag{13}$$

In equilibrium, Eq. (12) becomes a fixed point formulation

$$f_W \text{“ } \varphi\text{p}f_W \text{˚ } f_U\text{q,} \tag{14}$$

which needs to be solved for f_W, i.e., the fixed point is actually a pdf, what suggests using. This motivates solving the problem with a fixed point iteration. Starting with $f_W^{p0q} \text{“ } \delta_0$, Eq. (14) is applied as long as it converges. Since the repetitive convolution introduces a numerical accumulating error an additional exit condition based on the L^2 norm of the difference Δ^{piq} between

two consecutive iterations is added, such that the algorithm ends, if the iterations start to diverge. This approach is summarized in Algorithm 1.

Proving the convergence is out of scope of this paper, but the following explanation justifies the approach. In principle, the algorithm evaluates Lindley's equation again and again, which simulates the queue behavior, but in contrast to a simulation, the algorithm uses the pdf and thereby it evaluates all possible constellations at once, weighted by their probabilities.

F. Scheduling and Slicing

As already explained before, network slices are envisioned to treat the traffic from different applications. The switch model (c.f. Fig. 1(b)) already comprises multiple queues at each output for different services. To integrate this into the model, each queue Q_i is replaced by a system of queues [redacted] for each traffic class c P C. Each class may have its own arrival and service process, to reflect the characteristics of the considered traffic. In the following, a superscript pcq is added, whenever the variable refers to class c, and the superscript o is added to refer to the entire node.

For the sake of simplicity, only two classes C " t1,2u are considered for now. However, the approaches can be generalized for more classes. For class c, let c^{-}" 3´c denote the other class. Without loss of generality, let c " 1 be the class with the higher (or equal) priority. Further, the queue index i is omitted from now, since only single nodes are considered for the scheduling. In the following, different schemes to schedule the traffic from the priority queues in one ES along with the integration into the model are discussed.

1) Dedicated Resources: First, completely dedicated resources (DR) are considered, i.e., one slice can transmit r_1 P p0,1q of the time and the other slice gets r_2 " 1 ´ r_1 of the time resources, and within these constraints, both slices are scheduled independently. This results in effective service rates μ^{pcq} " $r_c\mu^{o}$ for queue Q^{pcq}, which will be taken to assess the waiting time of isolated queues.

2) Priority Queuing with preemption: If priority queuing is implemented, the queue with higher priority is always scheduled, when there is data to send. In contrast, low priority traffic can only be transmitted, when there is no traffic of higher priority. In priority queuing with preemption (PQwp), the processing of a lower priority queue will even be paused, as soon as higher priority traffic arrives, and not continued until the higher priority queue is empty again. In this case, the waiting time of the high priority class is not affected by the low priority, and so it simply results from the class 1 arrival process A^{p1q} and the service process of the overall node B^{o}.

The waiting time of class 2 is modeled as follows.by the following equation.

$$W_{\mathrm{PQwpp2q}} \text{ " } W_{\mathrm{o}} \text{ ` } \hat{W}_{\mathrm{p1q}} \text{ ´ } W_{\mathrm{PQwpp2q}} \text{ ` } S_{\mathrm{p2q}}^{-}. \quad (15)$$

Here, W^{o} is the waiting time, the node would have, if the traffic from both classes would be treated jointly without any priorities, since for the arriving object it does not matter in which order the elements that are in front are being served. In addition, the object experiences the additional waiting time $\hat{W}_{\mathrm{p1q}}\mathrm{p}t\mathrm{q}$ that arises due to the arriving high priority flows while waiting for a time t and will be discussed in Section II-F4. Eq. (15) is formulated implicitly as a fixed point formulation and can be solved iteratively again.

3) Priority Queuing without preemption: There is also priority queuing without preemption (PQwop). Here, the processing of a lower priority packet in service will always be finished, before the high priority queue is serviced again. Thus, high priority traffic only has to wait, if the queue was empty, and there is currently low priority traffic being served from the other queue. The waiting time of class c " 1, can now be calculated as the mixed RV

$$W_{\mathrm{PQwop}} \text{ " } \begin{cases} \#\delta_0, & \text{w.p. } \pi_0^{o}, \\ [\text{redacted}]_{\mathrm{aq0}}, & \text{w.p. } \pi_{0}^{-o}, \end{cases} \quad (16)$$

where π_0^{o} and π_0^{-o} are the probabilities of the entire node being empty and non-empty, respectively, and $W_x^{p1_a^q0}$ being the RV of the waiting time under the condition that the queue of class c is not empty. This is basically the waiting time occurring, when only class 1 would be served, but with the probabilities of no waiting for the overall traffic.

For the second class, the waiting time is similar to the case with preemption in Eq. (15), but the time parameter of $\hat{W}_{\mathrm{p1q}}$ is changed, because additional packets arriving during the service cannot interrupt the service.

$$W_{\mathrm{PQwopp2q}} \text{ " } W_{\mathrm{o}} \text{ ` } \hat{W}_{\mathrm{p1q}} \text{ ´ } W_{\mathrm{PQwopp2q}}^{-}. \quad (17)$$

4) Derivation of the Additional Waiting Time: It remains to derive the additional waiting time $\hat{W}$ ptq of a general queue, due to the new arrivals within time t. Therefore, let S_k and T_k be the RVs of the sum of k independent service times and inter-arrival times, respectively. Further, let A_t be the number of arrivals within time t, which distribution is given by

$$\Pr(A_t = k) = \Pr(A_t \geq k) - \Pr(A_t \geq k-1) \tag{18}$$

$$= F_{T_k}(t) - F_{T_{k-1}}(t). \tag{19}$$

For a fixed time t

$$\hat{W}(t) = \sum_{k=0}^{\infty} \Pr(A_t = k) S_k, \tag{20}$$

and thus, for a random time T this finally leads to

$$\hat{W}(T) = \sum_{k=0}^{\infty} \left(\int_{-\infty}^{\infty} f_T(t) \Pr(A_t = k)\, dt \right) S_k, \tag{21}$$

where the integral in parentheses calculates the probability of having k arrivals within the random time T.

The addends in Eqs. (15) and (17) are not independent, because they have the form $Z := \hat{W}(T) + T$. Thus, the accuracy can be improved by directly considering the distribution of Z instead.

$$f_Z(\xi) = F_Z'(\xi) = \frac{d}{dz} \Pr\left[W(T) + T \leq \xi\right],$$

$$= \frac{d}{dz} \int_{-\infty}^{\infty} \sum_{k=0}^{\infty} f_T(t) \Pr(A_t = k) \int_{-\infty}^{\xi - t} f_{S_k}(\omega)\, d\omega\, dt,$$

$$= \sum_{k=0}^{\infty} \left(\left(f_T(t) \cdot \Pr(A_t = k) \right) * f_{S_k}(\omega) \right)(\xi).$$

5) Weighted Fair Queuing: The last approach under investigation, weighted fair queuing (WFQ), lies in between DR and priority queuing. Here, a weighted round-robin is performed, by selecting the queue with the earliest virtual departure for the next transmission. As the virtual service rates are scaled by weighting factors w_c with $w_1 + w_2 = 1$, traffic from one class can be prioritized.

Since the general stochastic analysis of this scheduling approach is cumbersome, an approximation is derived in the following. For this purpose, limiting cases are considered. If the weights were set to the boundary values $(w_1, w_2) = (1, 0)$, WFQ would degenerate to PQwop, since in this case, class 1 and 2 would get infinite and zero as virtual service rates for the priority calculation, respectively, which translates to priority scheduling. If the weights were set equal, i.e., $(w_1, w_2) = (0.5, 0.5)$ or the regular round-robin, both queues have equal priority and so both classes experience the waiting time of the overall node W^{o}. This motivates the approximation by the following mixtures of RVs

$$W_{WFQ}^{(1)} = \begin{cases} W^{o}, & \text{w.p. } 1 - 4(w_1 - \frac{1}{2})^2 \\ W_{PQwop}^{(1)}, & \text{w.p. } 4(w_1 - \frac{1}{2})^2 \end{cases}, \tag{22}$$

$$W_{WFQ}^{(2)} = \begin{cases} W^{o}, & \text{w.p. } 1 - 4(w_2 - \frac{1}{2})^2 \\ W_{PQwop}^{(2)}, & \text{w.p. } 4(w_2 - \frac{1}{2})^2 \end{cases}. \tag{23}$$

Here, the probabilities were chosen in a way that the above mentioned limiting cases are satisfied, if (w_1, w_2) were chosen accordingly. Further, a quadratic term was chosen instead of a simple linear term, to take into account that the ratio [redacted] = [redacted] grows as $\frac{d}{dw_1}(1-w_1) = ($[redacted]$1-w_1)$, so that the PQwop models are favored.

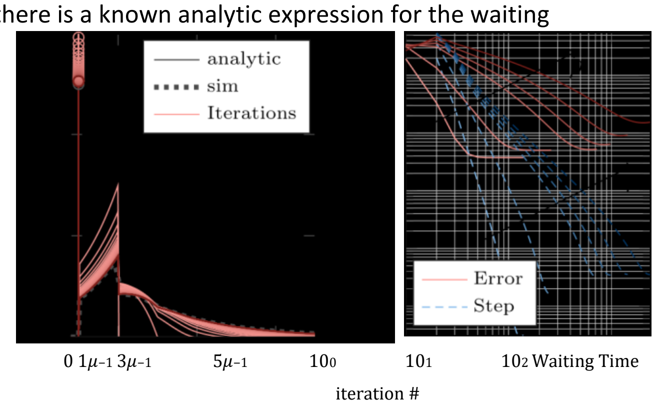
III. Numerical Evaluation and Validation

In this section, the model is evaluated and validated by comparison to simulation results. Therefore, the numerical approach for $GI/GI/1$ queues is evaluated first, before the accuracy and performance of the models for the different schedulers is assessed.

For the evaluation of the model, a class for general probability distributions was implemented. This class can hold continuous and discrete RV, but also mixtures of both kinds and thereby allows the necessary operations to perform easily.

A. Performance of the GI/GI/1 Approximation

The numerical approximation for the waiting time in general queues is tested with an $M/D/1$ queue for two reasons. Firstly, there is a known analytic expression for the waiting



(a) (b)

Fig. 2. Convergence of the pdfs of the waiting time distributions. (a) Illustration of the iterated pdfs. (b) For different values of the network load ρ, the approximation error compared to the analytic solution and the difference between consecutive iterations (step) are depicted with solid and dashed lines, respectively (both in L^2-norm).

time distribution for $M/D/1$ [11], such that it is possible to calculate the exact approximation error. Secondly, the $M/D/1$ queue has practical relevance in the considered scenario (c.f. Section II-C). The arrival and service rate were chosen as $\lambda = 0.15$ and $\mu = 0.2$, respectively, which corresponds to a load of $\rho = 0.75$.

The results are shown in Fig. 2. In (a), the pdfs of each iteration are depicted in different shades of red, where a darker red refers to a higher iteration index. As a result, the last iterations and the analytic solution are almost indistinguishable. Furthermore, the numerical solution resolves the discontinuities remarkably well and thereby outperforms the kernel-based pdf estimation from simulation results, which fails at discontinuities.

Fig. 2(b) shows the convergence speed with respect to the error in the L^2-norm for different values of the network load ρ

P t0.25,0.5,0.75,0.8,0.85,0.9u in solid red lines. Here, an error floor can be identified, due to the limitations of numerical convolution. Moreover, the difference (step) between two iterations L^2-norm, which is taken as a break condition, is depicted as well in dashed blue lines.

As expected, the convergence speed decreases for higher loads ρ, due to the convergence explanation in the end of Sec. II-E. However, since URLLC becomes unfeasible with higher loads, moderate load conditions are of higher interest.

B. Evaluation of the schedulers

For the evaluation of the scheduling, again $M\{D\{1$ models were chosen. However, to the best of our knowledge, there are no analytical results known for the different scheduling policies. Thus, the model is validated by comparison to simulation results. For the two traffic classes, equal arrival rates $\alpha^{p_c q}$ " 0.06 were chosen, and the overall service rate of the node was set to μ^o " 0.2, which refers to a moderate overall load of ρ^o " 0.6. It should be noted that the overall load cannot be too high, since this may result in instability of the low priority queues. The weights for the different high priority class in DR and WFQ are set to r_1 " w_1 " 0.6,

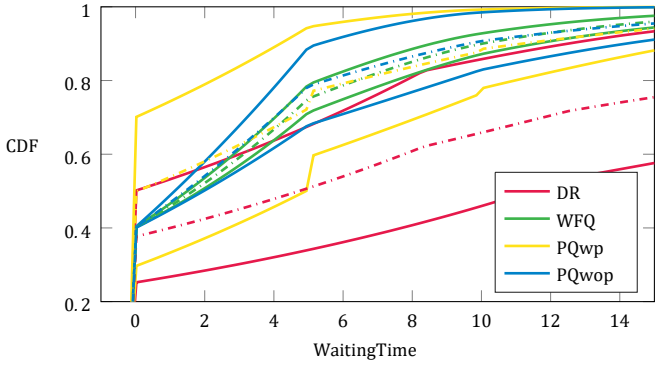


Fig. 4. The waiting time performance of the different scheduling policies. The solid lines refer to the different classes for each scheduler, whereas the dashed-dotted lines show the overall performance of both classes.

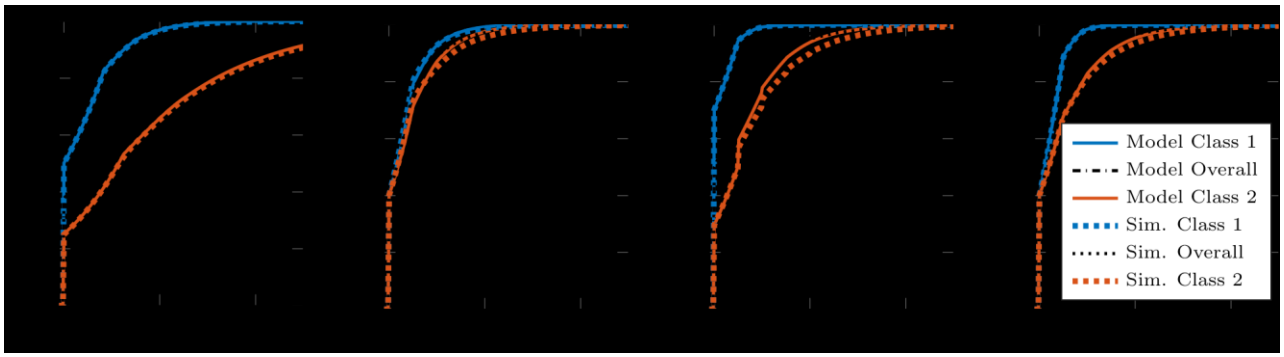
respectively. The results are depicted in Fig. 3, where the cdfs of the waiting times are shown for both classes and the overall node. It can be observed that the models manage to approximate the curves obtained from simulations well.

Furthermore, Fig. 4 aggregates the model curves in one plot in order to compare the performances of the different schedulers for the single classes (the two solid lines per class) as well as for the overall node (dashed dotted lines). Here, the inefficiency of slices with DR stands out, since the cdfs are located to the lower-right of the others. Even the higher priority slice performs only comparable to the lower priority of PQwop. Both priority queuing schedulers show similar overall performance, but the version with preemption clearly favors the high priority class. The performance of the different classes in WFQ form only a narrow corridor around the overall curve, which is also due to the chosen weights. However, by modifying the weights, the cdfs could be spread up to the limiting case of PQwop.

IV. Conclusion and Outlook

In the paper, modeling approaches for general queuing systems and for different scheduling policies for multiple traffic classes were presented. They proved to approximate simulation results well and are thus promising for the evaluation of real systems. One remarkable result is that the presented models not only provide the first or second moments, i.e., mean and variance, of important KPIs, such as latency, but offer the entire distribution in terms of the pdf or cdf. Thanks to that, percentiles as a measure of reliability can be easily calculated and so, guarantees like "p % of the packets have a latency below t" can be stated for any network. Thereby, it is a valuable tool for URLLC applications, where such guarantees are required. The model provides an accurate and flexible alternative to complex system and network simulations with extensive simulation times.

Hence, this work fills the gap on how to treat general queuing nodes in queuing networks for the evaluation of endto-end latency in mobile networks. For future work, the model is intended to be used for network optimization, e.g., through optimization of the scheduling parameters.



(a) (b) (c) (d)

Fig. 3. Evaluation of the performance of the modeling of the different scheduling policies, i.e., (a) Dedicated resources, (b) Weighted fair queuing, (c) Priority queuing with preemption, and (d) priority queuing without preemption. The legend in (d) applies to all subfigures.

References

[1] P. Schulz *et al.*, "Latency critical IoT applications in 5G: perspective on the design of radio interface and network architecture," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 70–78, February 2017.

[2] S. Schiessl, J. Gross, and H. Al-Zubaidy, "Delay analysis for wireless fading channels with finite blocklength channel coding," in *Proc. of the 18th ACM MSWiM '15*. New York, NY, USA: ACM, 2015, pp. 13–22.

[3] C. She, C. Yang, and T. Q. S. Quek, “Radio resource management for ultra-reliable and low-latency communications,” *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 72–78, 6 2017.

[4] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” *IEEE/ACM Trans. Netw.*, vol. 1, no. 4, pp. 397–413, Aug. 1993.

[5] M. Nagireddy, “Cisco Catalyst 9500 High Performance Switch Architecture,” July 2018, techWiseTV Workshop.

[6] 3GPP, “Study on RAN Improvements for Machine-type communications,” 3GPP, Tech. Rep. TR 37.868 V1.1.0, August 2011.

[7] M. Laner, P. Svoboda, N. Nikaein, and M. Rupp, “Traffic models for machine type communications,” in *ISWCS 2013; The Tenth International Symposium on Wireless Communication Systems*, Aug 2013, pp. 1–5.

[8] 3GPP, “Technical Specification Group Radio Access Network; Study on New Radio Access Technology; Physical Layer Aspects; (Release 14),” 3GPP, Tech. Rep. TR 38.802 V14.2.0, September 2017.

[9] D. Bertsekas and R. G. Gallager, *Data Networks*, 2nd ed. Englewood Cliffs, NJ: Prentice Hall, 1992.

[10] L. Kleinrock, *Queueing systems. 2, Computer applications*. New York: Wiley, 1976.

[11] G. Franx, “A simple solution for the M/D/c waiting time distribution,” *Operations Research Letters*, vol. 29, no. 5, pp. 221 – 229, 2001.