Latency and Peak Age of Information in Multipath Coded Communications

Federico Chiariotti, Beatriz Soret, Petar Popovski

***Abstract*—The use of parallel communication paths to provide reliable, low-latency service is a significant trend in cellular networks, as it can provide a way to satisfy the exacting Quality of Service (QoS) requirements of 5G-enabled applications. In particular, coding data across multiple paths can significantly improve reliability and reduce overall latency, compensating for stragglers and lost packets with the redundant information from other paths. However, the design trade-offs in optimizing these systems are non-trivial, particularly when considering Age of Information (AoI). In this work, we derive the latency and Peak Age of Information (PAoI) distributions for such a multipath coded system, drawing design insights on how to optimize either. While preemption is always the optimal choice to minimize AoI in a single-path, uncoded queuing system, the trade-off in this case is more complex, as dropping a late packet on one path might affect the reliability of the whole block. Our results show that the parameters to minimize the PAoI lead to poor latency performance, and optimizing both at once might require significant resource overprovisioning.**

I. Introduction

The use of parallel communication paths over different wireless technologies to provide additional reliability is one of the most interesting trends in 5G and Beyond networks, as it can help provide the stringent Quality of Service (QoS) requirements of the new classes of traffic defined by the network. In particular, packet duplication is a potential enabler for Ultra-Reliable Low-Latency Communications (URLLC) services [1] in cellular networks, and more advanced packet-level coding schemes can be used in end-to-end connection to provide high-throughput service with guarantees on the maximum latency [2]. More in general, the limitations of individual wireless paths, which can be affected by blockage and interference, can be overcome by combining multiple communication paths and technologies [3], sacrificing some capacity to greatly enhance reliability and latency.

The main theoretical tool to study these kinds of multipath systems is the fork-join queuing model [4], in which packets arrive synchronously to multiple queues. These systems are extremely general, and can represent complex networks with a series of parallel queues and even loops [5], but are also complicated to analyze, and most works limit themselves to the derivation of average performance [6], or bounds to the latency distribution [7]. In the same context, the recently developed concept of Age of Information (AoI) [8] has superseded latency in some contexts: as information freshness depends not only on the latency of each single packet, but on how much time passes between packet receptions, AoI is a more complete metric when considering process monitoring or control applications. Peak Age of Information (PAoI) [9] is a related metric often used when reliability is important, as it represents the maximum value of the AoI for each received packet.

However, AoI is still largely unexplored in fork-join systems, and there is a limited number of works considering it in this setting. The most general of these [10] deals with the average AoI in what we denote as $M/M/(K,N)$ systems, applying it in the context of distributed computing. Another work by Talak *et al.* [11] addresses the trade-off between AoI and latency, considering the possibility of choosing one or more paths with an intelligent scheduler. The paper shows that age-oriented systems will increase the latency for packets that do not contribute to information freshness (i.e., packets that arrive out of order), increasing both the average and the variance of the latency significantly.

In this paper, we consider a fork-join queue with deterministic arrivals and a Markovian service process, in which periodic blocks of data are encoded into N packets, only K of which are required to decode the whole blocks, and sent over N parallel links with exponentially distributed service times. With a slight abuse of the standard Kendall notation to describe queuing systems, we will refer to this model as $D/M/(K,N)$. We explicitly derive the latency and PAoI Probability Density Functions (PDFs) and Cumulative Density Functions (CDFs) for the cases with an infinite buffer and without buffer and a preemptive policy, denoted as $D/M/(K,N)/\infty$ and $D/M/(K,N)/1$, respectively, and analyze the trade-offs between latency and PAoI performance. We also include the more general $D/M/(K,N)/L$ case, which is analyzed by simulation. Our results show that the settings needed to minimize the PAoI lead to poor latency performance, and optimizing both at once might require significant resource overprovisioning.

The rest of the paper is organized as follows: first, the basic system model and notations used are described in Sec. II. The analysis for the $D/M/(K,N)/1$ case is presented in Sec. III, while the analysis for the $D/M/(K,N)/\infty$ case is presented in Sec. IV. The simulation settings and results are described in Sec. V, and Sec. VI concludes the paper and presents some possible avenues of future work.

Federico Chiariotti (email: fchi@es.aau.dk) and Petar Popovski (email: petarp@es.aau.dk) are with the Department of Electronic Systems, Aalborg University, Denmark. Beatriz Soret (email: bsa@es.aau.dk) is also with the Department of Communications Engineering, Universidad de Málaga, Spain. This work was partly funded by the IntelliIoT project under the H2020 framework grant ID 957218.

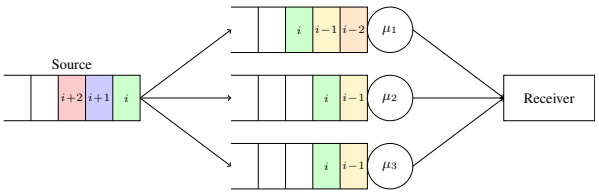


Fig. 1: Schematic of the system model with $N = 3$.

II. System model

We first introduce some notation. In the following, we use $p_X(x)$ to indicate the PDF of random variable X, and $P_X(x)$ to indicate its CDF. Random variables are denoted with capital letters, while values are lowecase. Vectors are in bold, e.g., $\mathbf{v}$, and matrices are bold and capitalized, e.g., $\mathbf{M}$.

We consider a parallel queuing system with N parallel queues and synchronized arrivals. A block of K packets is generated every τ seconds, and encoded using a packet erasure code into N packets. Each of the N packets is then transmitted over one of the queues. The block is decoded as soon as any set of K packets is correctly received.

Each individual system j has an exponentially distributed service time with rate μ_j, and a queue of (potentially infinite) size L: at any moment, there can only be up to L packets in each system. In this work, we consider a preemptive First Come First Serve (FCFS) queuing policy, so that the packet currently in service is dropped if a new packet is generated and finds L packets ahead of it. Packet dropping is performed independently on each individual system. Therefore, if some packets are dropped on one system, the blocks they belong to might still be decoded correctly if a sufficient number of packets is correctly received on the other systems. Additionally, each system has a packet erasure probability ε_j, so that every transmitted packet might be undecodable at the receiver due to channel impairments.

If block i is generated at time g_i and its packet is received on system j at time $r_{i,j}$, with $r_{i,j} = \infty$ if the packet is dropped or erased, we can compute the delivery latency D_i:

$$D_i = \inf\left(t \in \mathbb{R} : \sum_{j=1}^{N} \mathbb{1}(t - r_{i,j}) \geq K\right), \tag{1}$$

where $\mathbb{1}(x)$ is the step function, which is equal to 1 if $x \geq 0$ and 0 otherwise. Naturally, some blocks might not be delivered at all, as there might be more than $N - K$ erased or dropped packets. As such, the CDF of the latency does not reach 1, except for the error-free case with $L = \infty$, in which no block is lost. We can also define the AoI $\theta(t)$ as the time elapsed since the generation of the last correctly received block:

$$\theta(t) = t - \sup\left(g_i \in \mathbb{R} : g_i + D_i \leq t\right). \tag{2}$$

We can then define the PAoI Δ_i, which is the AoI measured at the instant right before decoding for the i-th packet:

$$\Delta_i = \theta(g_i + D_i - \epsilon) + \epsilon, \tag{3}$$

where ϵ is an arbitrarily small positive quantity.

III. Analysis: coded $D/M/(K,N)/1$ system with preemption

We can first consider each system to apply the FCFS queuing policy over a preemptive system, which has been shown to be optimal in the $K = N = 1$ case, in which each new set of packets goes into service immediately, regardless of previous packets. The PDF $p_{D_{i,j}}(t)$ of the delivery time on path j is then exponentially distributed:

$$p^{(1)}_{D_{i,j}}(t) = (1 - \varepsilon_j)\mu_j e^{-\mu_j t}. \tag{4}$$

Naturally, the corresponding CDF $P^{(1)}_{D_{i,j}}(t)$ is given by:

$$P^{(1)}_{D_{i,j}}(t) = \int_0^t p^{(1)}_{D_{i,j}}(x)dx = (1 - \varepsilon_j)\left(1 - e^{-\mu_j t}\right). \tag{5}$$

As we remarked above, this PDF does not sum to 1, as the latency is considered as infinite if there is an erasure, which occurs with probability ε_j. We first denote the set of numbers from 1 to N as $\mathcal{N} = \{1, \ldots, N\}$. We then define $\mathcal{S}(M, \mathcal{N})$ as the set of possible unordered sets of non-repeating indices of length M:

$$\mathcal{S}(M, \mathcal{N}) = \left\{\mathcal{L} \in \mathcal{N}^M : i \neq j \,\forall i, j \in \mathcal{L}\right\}. \tag{6}$$

The block will be decoded whenever one of these sets of paths delivers the packet without erasure. This means that the decoding happens in instant t if $K - 1$ packets have already been delivered successfully, $N-K-2$ have not been delivered or have been erased, and the final packet is delivered exactly at time t. The PDF of the delivery latency is then given by:

$$\begin{aligned} p^{(1)}_{D_i}(t) = &\sum_{\mathcal{L}\in\mathcal{S}(K-1,\mathcal{N})} \prod_{j\in\mathcal{L}} (1 - \varepsilon_j)\left(1 - e^{-\mu_j t}\right) \sum_{\ell\in\mathcal{N}\setminus\mathcal{L}} (1 - \varepsilon_\ell) \\ &\times \mu_\ell e^{-\mu_\ell t} \prod_{m\in\mathcal{N}\setminus\mathcal{L}\setminus\{\ell\}} \left(\varepsilon_m + (1 - \varepsilon_m)e^{-\mu_m t}\right). \end{aligned} \tag{7}$$

In this case, the packets in set $\mathcal{L}$ have already been delivered by time t, while packet ℓ is delivered exactly at time t, and the remaining packets are erased or undelivered. We can then compute the corresponding CDF:

$$\begin{aligned} P^{(1)}_{D_i}(t) &= \int_0^t p^{(1)}_{D_i}(x)dx \\ &= \sum_{M=K}^{N} \sum_{\mathcal{L}\in\mathcal{S}(M,\mathcal{N})} \prod_{j\in\mathcal{L}} P^{(1)}_{D_{i,j}}(t) \prod_{\ell\in\mathcal{N}\setminus\mathcal{L}} \left(1 - P^{(1)}_{D_{i,\ell}}(t)\right) \\ &= \sum_{M=K}^{N} \sum_{\mathcal{L}\in\mathcal{S}(M,\mathcal{N})} \prod_{j\in\mathcal{L}} (1 - \varepsilon_j)\left(1 - e^{-\mu_j t}\right) \\ &\quad \times \prod_{\ell\in\mathcal{N}\setminus\mathcal{L}} \left(\varepsilon_\ell + (1 - \varepsilon_\ell)e^{-\mu_\ell t}\right). \end{aligned} \tag{8}$$

The success probability for decoding a block is simply given by $p^{(1)}_s = P^{(1)}_{D_i}(\tau)$, as each packet is always in the first position in the queue and immediately enters service, but it is dropped as soon as the next block of data is generated. As failures are independent, the number of consecutive failed transmissions

follows a geometric distribution, and we can simply get the PDF of the PAoI as:

$$p^{(1)}_{\Delta_i}(\xi) = (1-p^{(1)}_s)^{\max\left(\lfloor \frac{\xi}{\tau} \rfloor -1,0\right)} p^{(1)}_{D_i}(\text{mod}(\xi,\tau)). \quad (9)$$

We can also derive the PAoI CDF:

$$P^{(1)}_{\Delta_i}(\xi) = (p^{(1)}_s)^{\lfloor \frac{\xi}{\tau} \rfloor -1} + (1-p^{(1)}_s)^{\lfloor \frac{\xi}{\tau} \rfloor -1} P^{(1)}_{D_i}(\text{mod}(\xi,\tau)). \quad (10)$$

The complexity of computing the delivery latency PDF and CDF is considerable, as there are several nested sums over subsets. The complexity of the PDF computation is $O\left(\frac{N!}{(K-2)!(N-K-1)!}\right)$, as it depends on iterating over the subsets of size $K-1$ of the set of packets, and the complexity for each cycle is $O((N-K)(N-K+1)(K-1))$. The CDF computation iterates over subsets of size between K and N, and each cycle requires $O(N)$ operations. The overall complexity is then $O\left({}_2F_1(1,K-N,1+K;-1)\binom{N}{K}N\right)$, where ${}_2F_1(a,b,c;x)$ is the ordinary hypergeometric function. On the other hand, the PAoI computation is relatively simple, once the latency distribution is known.

IV. Analysis: coded $D/M/(K,N)/\infty$ system

We can now consider a classical queuing system with infinite buffers and FCFS queuing. The steady-state probability right before a new arrival was derived in [12] using Palm probability theory [13], and is given by:

$$p^{(\infty)}_{Q_{i,j}}(q_{i,j}) = (1-\sigma_j)\sigma_j^{q_{i,j}}, \quad (11)$$

where the parameter σ_j is the solution in $(0,1)$ to the following equation:

$$x = e^{2\mu_j\tau(x-1)}. \quad (12)$$

We know that the delivery time for packet i, which finds $q_{i,j}$ packets queued ahead of it in system j, follows an Erlang distribution [14], as the packet is never dropped:

$$p^{(\infty)}_{D_{i,j}|Q_{i,j}}(t|q_{i,j}) = (1-\varepsilon_j)\frac{(\mu_j t)^{q_{i,j}} e^{-\mu_j t}}{q_{i,j}!}. \quad (13)$$

The corresponding CDF $P^{(\infty)}_{D_{i,j}|Q_{i,j}}(t|q_{i,j})$ is given by:

$$P^{(\infty)}_{D_{i,j}|Q_{i,j}}(t|q_{i.j}) = (1-\varepsilon_j)\left(1-\sum_{n=0}^{q_{i,j}}\frac{(\mu_j t)^n e^{-\mu_j t}}{n!}\right). \quad (14)$$

Note that this is not just the Erlang CDF, but also has a $(1-\varepsilon_j)$ term that accounts for the possibility of erasure. In order for the transmission of the coded block to be successful, the packets from the links contained in one of the vectors in $\mathcal{S}(K,\mathcal{N})$ must be delivered correctly. We can then express the probability of decoding the block after latency t for a given state $\mathbf{Q}_i$ as the product of the probability of having decoded $K-1$ packets beforehand and getting the K-th at time t:

$$p^{(\infty)}_{D_i|\mathbf{Q}_i}(t|\mathbf{q}_i) = \sum_{\mathcal{L}\in\mathcal{S}(K-1,\mathcal{N})}\prod_{j\in\mathcal{L}} P^{(\infty)}_{D_{i,j}|Q_{i,j}}(t|q_{i,j}) \times \sum_{\ell\in\mathcal{N}\setminus\mathcal{L}} p^{(\infty)}_{D_{i,\ell}|Q_{i,\ell}}(t|q_{i,\ell}) \prod_{m\in\mathcal{N}\setminus\mathcal{L}\setminus\{\ell\}}\left(1-P^{(\infty)}_{D_{i,m}|Q_{i,m}}(t|q_{i,m})\right). \quad (15)$$

Like in the $D/M/(K,N)/1$ case, the set $\mathcal{L}$ contains the $K-1$ paths that successfully transmitted their packets before t, while the packet on path ℓ is received exactly at time t. The remaining paths either have an erasure event or still have to deliver their packet. As the systems are separable, the steady-state probability of being in a given state for the j-th system is independent from all others, and follows (11). We can then remove the condition on (13) to get $p_{D_{i,j}}(t)$:

$$\begin{aligned} p^{(\infty)}_{D_{i,j}}(t) &= \sum_{q_{i,j}=0}^{\infty}\pi_j(q_{i,j}) p^{(\infty)}_{D_{i,j}|Q_{i,j}}(t|q_{i.j}) \\ &= \sum_{q_{i,j}=0}^{\infty}(1-\varepsilon_j)(1-\sigma_j)\sigma_j^{q_{i,j}}\mu_j\frac{(\mu_j t)^{q_{i,j}}e^{-\mu_j t}}{q_{i,j}!} \\ &= (1-\varepsilon_j)(1-\sigma_j)\mu_j e^{-\mu_j(1-\sigma_j)t}. \end{aligned} \quad (16)$$

In the same way, we remove the condition on (14) for each individual system:

$$\begin{aligned} P^{(\infty)}_{D_{i,j}}(t) &= \sum_{q_{i,j}=0}^{\infty}\pi_j(q_{i,j}) P^{(\infty)}_{D_{i,j}|Q_{i,j}}(t|q_{i.j}) \\ &= \sum_{q_{i,j}=0}^{\infty}(1-\varepsilon_j)(1-\sigma_j)\sigma_j^{q_{i,j}}\left(1-\sum_{n=0}^{q_{i,j}}\frac{(\mu_j t)^n e^{-\mu_j t}}{n!}\right) \\ &= (1-\varepsilon_j)\left(1-e^{-\mu_j(1-\sigma_j)t}\right). \end{aligned} \quad (17)$$

We can now substitute (17) and (16) into (15) to get the latency PDF for the block:

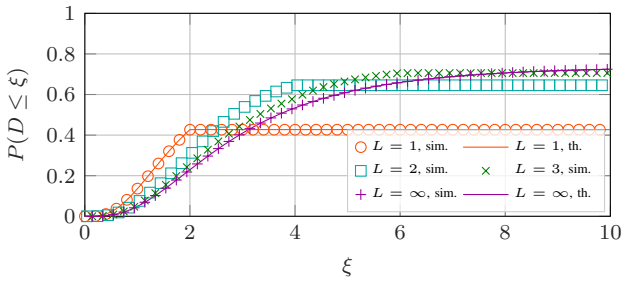
$$\begin{aligned} p^{(\infty)}_{D_i}(t) = &\sum_{\mathcal{L}\in\mathcal{S}(K-1,\mathcal{N})}\prod_{j\in\mathcal{L}}(1-\varepsilon_j)\left(1-e^{-\mu_j(1-\sigma_j)t}\right) \\ &\times\sum_{\ell\in\mathcal{N}\setminus\mathcal{L}}(1-\varepsilon_\ell)(1-\sigma_\ell)\mu_\ell e^{-\mu_\ell(1-\sigma_\ell)t} \\ &\times\prod_{m\in\mathcal{N}\setminus\mathcal{L}\setminus\{\ell\}}\left(\varepsilon_m+(1-\varepsilon_m)e^{-\mu_j(1-\sigma_j)t}\right). \end{aligned} \quad (18)$$

We can easily compute the corresponding CDF:

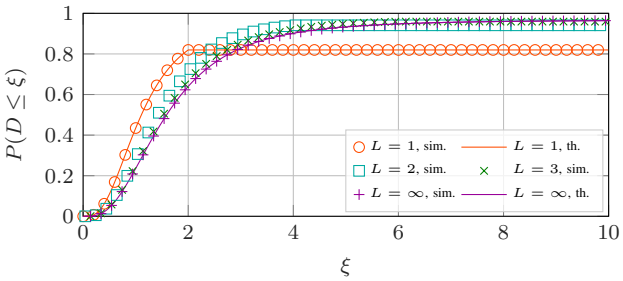
$$\begin{aligned} P^{(\infty)}_{D_i}(t) &= \int_0^t p^{(\infty)}_{D_i}(x)dx \\ &= \sum_{M=K}^{N}\sum_{\mathcal{L}\in\mathcal{S}(M,\mathcal{N})}\prod_{j\in\mathcal{L}}P^{(\infty)}_{D_{i,j}}(t)\prod_{\ell\in\mathcal{N}\setminus\mathcal{L}}\left(1-P^{(\infty)}_{D_{i,\ell}}(t)\right) \\ &= \sum_{M=K}^{N}\sum_{\mathcal{L}\in\mathcal{S}(M,\mathcal{N})}\prod_{j\in\mathcal{L}}(1-\varepsilon_j)\left(1-e^{-\mu_j(1-\sigma_j)t}\right) \\ &\quad\times\prod_{\ell\in\mathcal{N}\setminus\mathcal{L}}\left(\varepsilon_\ell+(1-\varepsilon_\ell)e^{-\mu_\ell(1-\sigma_\ell)t}\right). \end{aligned} \quad (19)$$

As no packets are dropped from the queue, the success probability p_s for a block is given by:

$$p^{(\infty)}_s = \sum_{m=K}^{N}\sum_{\mathcal{L}\in\mathcal{S}(M,\mathcal{N})}\prod_{j\in\mathcal{L}}\prod_{\ell\in\mathcal{N}\setminus\mathcal{L}}(1-\varepsilon_j)\varepsilon_\ell. \quad (20)$$



(a) Latency CDF for the $(4, 5)$ system.



(b) Latency CDF for the $(4, 7)$ system.

Fig. 2: Latency CDF for different queue sizes and codes with $\tau = 2$ and $\varepsilon = 0.2$.

In order to get the latency distribution for successful blocks, i.e., conditioning on the block's success, it is sufficient to divide $p_{D_i}^{(\infty)}(t)$ by $p_s^{(\infty)}$. We can then compute the PAoI PDF, considering that failures are independent:

$$p_{\Delta_i}^{(\infty)}(\xi) = \sum_{e=0}^{\left\lfloor \frac{\xi}{\tau} \right\rfloor} (1 - p_s^{(\infty)})^e p_{D_i}^{(\infty)}(\xi - e\tau). \tag{21}$$

The PAoI CDF calculation is also straightforward:

$$P_{\Delta_i}^{(\infty)}(\xi) = \sum_{e=0}^{\left\lfloor \frac{\xi}{\tau} \right\rfloor} (1 - p_s^{(\infty)})^e P_{D_i}^{(\infty)}(\xi - e\tau). \tag{22}$$

The computational complexity of the latency distribution calculations is the same as for the case with $L = 1$, but the PAoI PDF calculation is more complex, as it requires a sum. Its complexity is then $O\left(\frac{\xi}{\tau}\right)$, after the latency PDF and CDF have already been computed.

V. Simulation Settings and Results

In the following, we verify our analytical calculations by comparing them with the results of Monte Carlo simulations on the system. The simulations were run for $N_p = 10^6$ packets in each case, and the empirical and analytical CDFs match perfectly.

We consider a challenging scenario in which 3 slow paths have $\mu_s = 0.75$, while all the others have $\mu_f = 1$, e.g., $\boldsymbol{\mu} = (0.75, 0.75, 0.75, 1, 1)$ if $N = 5$. A shorter queue can represent both an advantage and a disadvantage if the load on the system changes: if there are enough redundant paths, systems with a lower L can deal with a higher load by dropping straggling packets, but this can quickly lead to a far lower decoding probability if the redundancy is too low. Fig. 2 shows the CDF of the latency for this scenario in the case where $\tau = 2$. We can see that the difference between a system with $L = 3$ and one with $L = \infty$ is limited, as there are rarely more than 3 packets in the queue, and they will usually be i nthe slower queues. On the other hand, the effect of having a shorter queue is noticeable: in the case with $L = 1$, all delivered packets arrive before $\tau = 2$, but almost 60% of blocks in the $(4, 5)$ system are lost due to packet errors or queue dropping. Even in the $(4, 7$ system, which has a significant amount of redundancy (and, if none of the faster paths have erasures, can deliver the block without the packets from the slower paths), almost

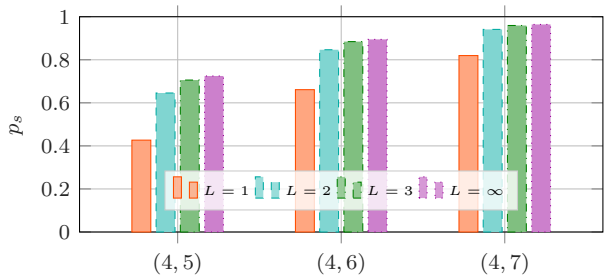
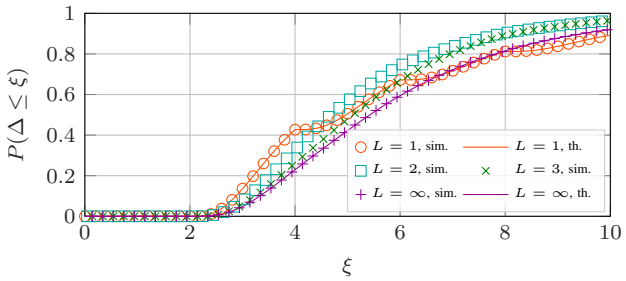


Fig. 3: Reliability for different amounts of redundancy and queue length configurations with $K = 4$, $\varepsilon = 0.2$, and $\tau = 1.5$.

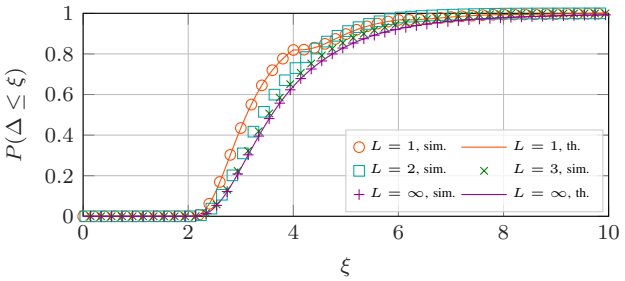
20% of blocks are lost. As expected, systems with a longer maximum queue size have a higher reliability, but this comes at the cost of a higher average latency, as packets might find longer queues ahead of them.

We can analyze the system reliability further by looking at Fig. 3, which shows the block decoding success probability for each configuration. The difference between $L = 3$ and $L = \infty$ is very small, and decreases as we add more redundancy, as dropped packets can be compensated for by other paths, while the difference between $L = 2$ and $L = 3$ already becomes noticeable. The case with $L = 1$ is an extreme example of this, as the dropping probability is $e^{-\mu_s \tau} \simeq 0.22$: if we combine this with the erasure probability for the packets that do manage to get transmitted before the next block generation instant, it is easy to understand why systems with $L = 1$ require a very high level of redundancy.

If we consider the PAoI, an interesting trade-off emerges, as shown in Fig. 4: as long as the redundancy is high enough, the system with $L = 1$ has a lower average and tail PAoI, and the system with $L = \infty$ performs significantly worse. On the other hand, $L = 2$ is the optimal choice in the $(4, 5)$ system, as the high block error rate increases the age for $L = 1$, and even makes the tail of the distribution worse than the $L = \infty$. In general, while finding a balance between latency and block error rate can be achieved, extremely short queues are highly beneficial for PAoI, but incur a cost in terms of reliability. Setting the appropriate queue length is a non-trivial optimization, particularly when redundancy is limited: this is a stark difference from the AoI optimization on simpler queuing networks with Markovian service, in which preemption is always the best choice. However, the general performance



(a) PAoI CDF for the $(4, 5)$ slow connection.



(b) PAoI CDF for the $(4, 7)$ slow connection.

Fig. 4: PAoI CDF for different queue sizes and codes with $\tau = 2$ and $\varepsilon = 0.2$.

trend still fits the conclusions from [11]: while $L = \infty$ and a lower load are the best choices if the system is aimed at having a high reliability, there is an inevitable trade-off with the PAoI and average latency. In the same way, optimizing the PAoI will lead to a relatively high block loss, as frequent preemption remains an effective method to minimize the age.

We can then look at the optimization of the redundancy level in our system: we consider a fixed number of paths $N = 6$, all of which have $\varepsilon = 0.1$ and $\mu = 1$: in this case, K is not fixed, but we consider a data block payload with a set size M. We can then choose the value of K, and consequently the amount of redundancy in the multipath transmission. However, as the payload size is fixed, adding more redundancy means transmitting bigger packets on each path and increasing the system load. This results in a multiplication of the service time by a factor $\frac{M}{K}$. The offered traffic G, which corresponds to the system load on the slowest path if the blocks are transmitted without any redundancy, as an independent variable:

$$G = \frac{M}{N\tau \min_{j\in\{1,\dots,N\}} \mu_j}. \tag{23}$$

We then examine the latency of the various systems for $M = 4.5$, fixing $\tau = 1.5$, which corresponds to an offered traffic $G = 0.5$. The left side of Fig. 5 shows the CDF of the latency for different queue lengths and codes. Interestingly, the best choice for $L = 1$, as shown in Fig. 5a, is to set $K = 1$ and maximize redundancy. The system can remain stable thanks to preemption, and only one packet is needed to recover the whole block, which increases the probability of the block getting transmitted on time. However, the system still runs into the reliability problems we discussed above due to excessive dropping. On the other hand, the systems with $L = 2$ and $L = \infty$, whose latency CDFs are shown in Fig. 5c and Fig. 5e, respectively, tend to balance redundancy and reliability. The system with $L = 2$ performs best when there is enough redundancy to protect the transmission, but not so much that packets are dropped too frequently: in this case, the best setting is $K = 3$. On the other hand, the best choice for the $L = \infty$ system seems to be $K = 4$, but lower values of K lead to instability, as packets are never dropped and the load on the paths becomes higher than 1.

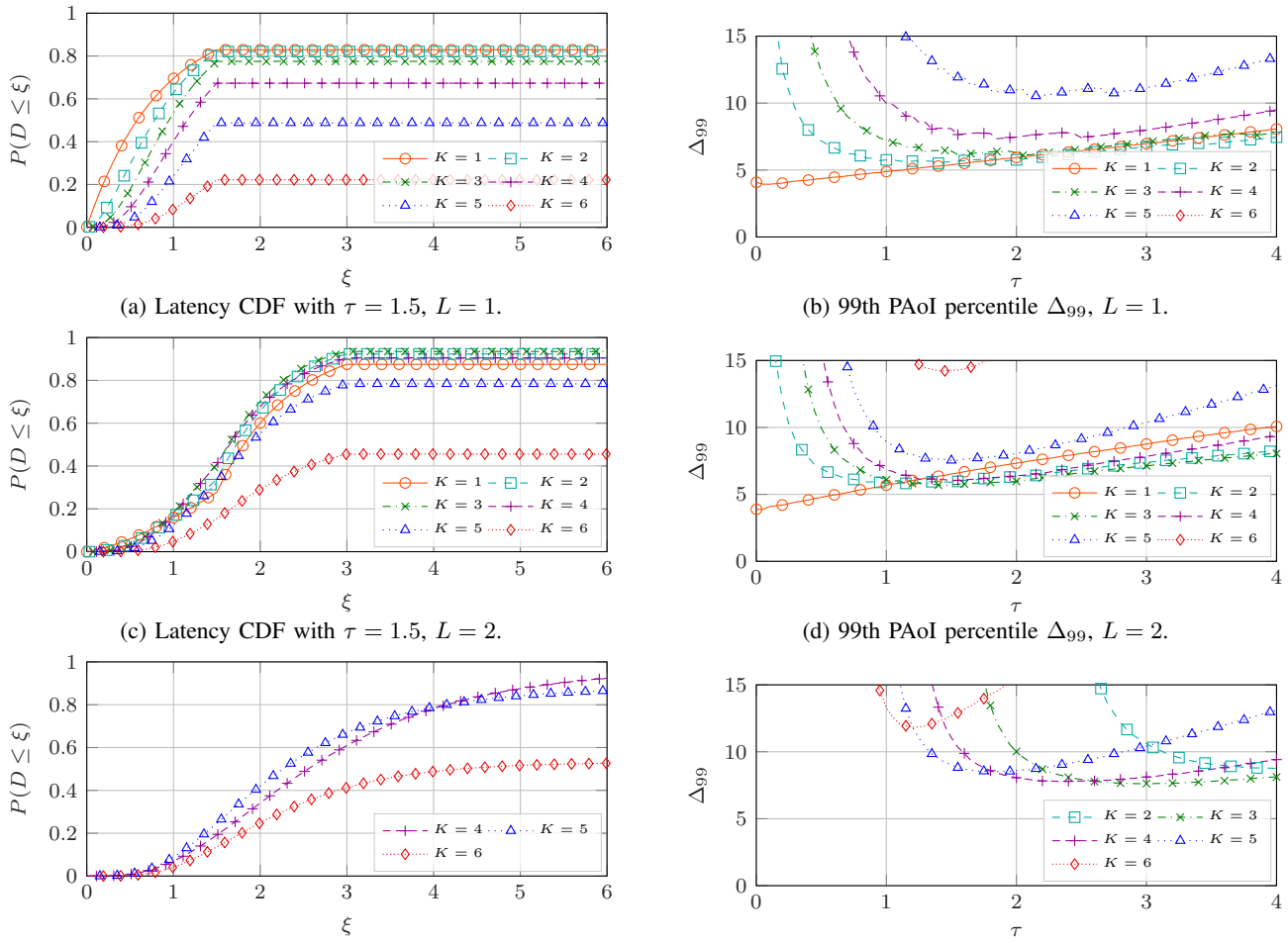
In general, the optimal choice when considering latency seems to be exploiting redundancy as much as possible without overloading the system: the fewer packets need to get through, the higher the probability that they will before they are dropped. In cases where the offered traffic is already high, as for $G = 0.5$, strict preemption becomes too aggressive, and allowing a short queue such as $L = 2$ or $L = 3$ is the best option. In almost every case, setting $L = \infty$ is not a good choice, as queuing delay becomes an important factor in increasing the latency. However, it provides the best reliability when the offered traffic is low, as no packets are dropped, and is the best choice in connections with very high erasure rates.

Increasing redundancy seems to be the optimal choice to minimize the PAoI as well, as the right side of Fig. 5 shows: we considered a fixed block size $M = 4.5$ (corresponding to $G = 0.5$ with $\tau = 1.5$), and plotted the 99th percentile Δ_{99} of the PAoI as a function of τ. Setting $K = 1$ and a very low τ is clearly the optimal choice to minimize the PAoI, even though it leads to a very low reliability. This result holds for both $M = 2.25$ and $M = 4.5$, as well as for $L = 1$ and $L = 2$. In the case with $L = \infty$, the stability of the queue becomes a concern, and the optimal setting is actually $K = 3$, but this system has a significant disadvantage in terms of PAoI with respect to the ones with finite, short queues. Interestingly, the results from uncoded transmission hold, as exploiting preemption seems to be the best choice. Furthermore, the trade-off between age and latency-reliability described by Talak *et al.* [11] is clearly still crucial: setting $L = 1$ and $\tau = 0.05$, as would be optimal with $M = 4.5$, results in $\Delta_{99} \simeq 4$ and a reliability of 25%.

VI. Conclusion

In this paper, we have analyzed the $D/M/(K, N)$ fork-join queue with packet-level coding, deriving the latency and PAoI distributions for $L = 1$ with preemption and $L = \infty$ with FCFS queuing. We also derived the PDF of the PAoI for $L \in \{1, 2, \infty\}$. These analytical results may be useful in the future development of the field, as well as to inspire system design in distributed computing and multipath communication.

Our results show that maintaining a short queue can be beneficial to the system in terms of both latency and PAoI, as strict preemption can make the decoding of data blocks more difficult by dropping packets too frequently. Having a longer queue is beneficial when K is close to N: in those cases, packets critical to decode the block end up being dropped. The optimization of the queue length with respect to the expected load is less trivial, as it involves a trade-off between not dropping packets too often and maintaining shorter queues.



(a) Latency CDF with $\tau = 1.5$, $L = 1$.

(b) 99th PAoI percentile Δ_{99}, $L = 1$.

(c) Latency CDF with $\tau = 1.5$, $L = 2$.

(d) 99th PAoI percentile Δ_{99}, $L = 2$.

(e) Latency CDF with $\tau = 1.5$, $L = \infty$.

(f) 99th PAoI percentile Δ_{99}, $L = \infty$.

Fig. 5: Latency and PAoI performance for different coding schemes with $N = 6$, $M = 4.5$, $\mu = 1$ and $\varepsilon = 0.1$.

Future work on the subject involves the investigation of the general case with $L > 1$, as well as more practical models oriented at multipath communication. In particular, high-throughput real-time applications such as Augmented Reality (AR) and Virtual Reality (VR) are an interesting application of these models, and integrating traffic models for them into the framework will be an interesting development for 5G and beyond.

References

[1] J. J. Nielsen, R. Liu, and P. Popovski, "Ultra-reliable low latency communication using interface diversity," *IEEE Transactions on Communications*, vol. 66, no. 3, pp. 1322–1334, Mar. 2018.

[2] F. Chiariotti, S. Kucera, A. Zanella, and H. Claussen, "Analysis and design of a latency control protocol for multi-path data delivery with pre-defined QoS guarantees," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1165–1178, Jun. 2019.

[3] M.-T. Suer, C. Thein, H. Tchouankem, and L. Wolf, "Multi-connectivity as an enabler for reliable low latency communications—an overview," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 156–169, Jan. 2020.

[4] C. Kim and A. K. Agrawala, "Analysis of the fork-join queue," *IEEE Transactions on Computers*, vol. 38, no. 2, pp. 250–255, Feb. 1989.

[5] Y. Dallery, Z. Liu, and D. Towsley, "Properties of fork/join queueing networks with blocking under various operating mechanisms," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 4, pp. 503–518, Aug. 1997.

[6] W. R. KhudaBukhsh, A. Rizk, A. Frömmgen, and H. Koeppl, "Optimizing stochastic scheduling in fork-join queueing models: Bounds and applications," in *Conference on Computer Communications (INFOCOM)*. IEEE, May 2017, pp. 1–9.

[7] A. Rizk, F. Poloczek, and F. Ciucu, "Stochastic bounds in fork–join queueing systems under full and partial mapping," *Queueing Systems*, vol. 83, no. 3, pp. 261–291, Aug. 2016.

[8] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *International Conference on Computer Communications (INFOCOM)*. IEEE, Mar. 2012, pp. 2731–2735.

[9] R. Devassy, G. Durisi, G. C. Ferrante, O. Simeone, and E. Uysal, "Reliable transmission of short packets through queues and noisy channels under latency and peak-age violation guarantees," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 4, pp. 721–734, Feb. 2019.

[10] B. Buyukates and S. Ulukus, "Timely distributed computation with stragglers," *IEEE Transactions on Communications*, vol. 68, no. 9, pp. 5273–5282, Jun. 2020.

[11] R. Talak and E. H. Modiano, "Age-delay tradeoffs in queueing systems," *IEEE Transactions on Information Theory*, vol. 67, no. 3, pp. 1743–1758, Mar. 2021.

[12] D. Pinotsi and M. A. Zazanis, "Synchronized queues with deterministic arrivals," *Operations Research Letters*, vol. 33, no. 6, pp. 560–566, Nov. 2005.

[13] F. Baccelli and P. Brémaud, *Palm probabilities and stationary queues*, ser. Lecture Notes in Statistics. Springer Verlag, Dec. 2012, vol. 41.

[14] A. K. Erlang, "Løsning af nogle problemer fra sandsynlighedsregningen af betydning for de automatiske telefoncentraler," *Elektroteknikeren*, vol. 13, pp. 5–13, Jan. 1917.