This document is a preprint of: N. Schwarzenberg, A. Trassl, F. Burmeister et al., “Channel-Aware Multi-User Resource Allocation for Ultra-Reliable Low-Latency Communications,” in Proceedings of IEEE International Symposium On Personal, Indoor And Mobile Radio Communications (PIMRC 2023), Toronto, Canada, Sep 2023.

Channel-Aware Multi-User Resource Allocation for Ultra-Reliable Low-Latency Communications

Nick Schwarzenberg, Andreas Traßl, Friedrich Burmeister, Richard Jacob and Gerhard Fettweis
Vodafone Chair Mobile Communications Systems, Technische Universität Dresden, Germany
{nick.schwarzenberg, andreas.trassl, friedrich.burmeister, richard.jacob, gerhard.fettweis}@tu-dresden.de

***Abstract*—Achieving high reliability in the presence of fading is particularly challenging under latency constraints, because the usual way of error mitigation by repetition becomes unfavorable. On the other hand, multi-connectivity does improve reliability without adding latency, but multiplies the required bandwidth per link and does not scale to a large number of users. There is hence a need for frequency diversity in a spectrum-efficient way. In this work, we investigate multi-user resource allocation schemes both without and with knowledge of each user's channel state. We evaluate the allocation-dependent reliability in terms of outage rate and outage duration based on simulations of automated guided vehicles in an industrial environment. To increase validity and ensure real-world correlation between vehicles, we draw channel states from high-resolution channel measurements at a factory floor. For channel-aware allocation, we propose a near-optimal low-complexity algorithm using different quality functions based on channel state preference lists. Since accurate channel information per user and resource incurs signaling overhead, we also evaluate the algorithm's sensitivity to the number and bandwidth of resources as well as to outdated channel information. In conclusion, channel-aware allocation offers significant reliability improvements over static allocation and emerges as a key enabler to realize ultra-reliable low-latency communications on a larger scale.**

***Index Terms*—Radio Resource Allocation, Scheduling, Reliability, Diversity, Industrial Radio, Industry 4.0, IIoT, URLLC.**

I. Introduction

Industrial use cases such as remote control of mobile robots and wireless safety require Ultra-Reliable Low-Latency Communications (URLLC) for uninterrupted operation and fast reaction times. However, providing URLLC over fading radio channels that are commonly found in factory halls is challenging. The usual approach to mitigate occasional errors is to rely on time diversity and repeat the lost data, but this comes at the cost of increased latency. On the other hand, leveraging frequency diversity in terms of Multi-Connectivity (MC) improves reliability without additional latency. MC is being implemented in 5G networks as packet duplication [1] and in Wi-Fi 7 as multi-link operation [2]. However, MC multiplies the required bandwidth per link and does not scale to many users.

Consider an industrial automation use case with an assembly line consisting of multiple remote-controlled mobile robots. Each robot periodically reports sensor data to a centralized controller and receives control signals in return, forming a closed loop. Assuming that the robots work jointly on a task which requires each of them to stay operational and connected, it follows that any communication outage at any robot causes a failure of the task. For each robot, the channel quality of its allocated radio resources must satisfy at all times the requirements of the wireless communication system for correctly transmitting the given data within a certain time (e.g., the duration of a control cycle). The reliability of the assembly line is hence limited by the robot with the worst channel conditions, and the resource allocation is optimal if it maximizes the channel quality of the weakest user.

The optimization criterion of maximizing the minimum rate is formally known as max-min fairness or max-min allocation [3]. It is however not trivial to find an optimal allocation for the max-min criterion. The authors in [4] choose a slightly different goal and resort to game theory to find fair allocations in a multi-user multi-carrier system. In [5], Traßl et al. propose an algorithm to find the optimal max-min allocation, but it is only evaluated for small numbers of users and resources. The main author suggests that the algorithm does not scale to the system size considered in this work because of computational complexity. Besides, the authors consider only Rayleigh fading with no correlation between the resources. The goal of this work is hence to find a computationally feasible allocation algorithm that follows the max-min criterion, and to evaluate its performance using real-world channel data.

II. System Model

A. Wireless Topology

We assume a set of users (e.g., mobile robots) $\mathcal{U} = \{1, 2, \ldots, U\}$ that are served by one Base Station (BS). For this work, we focus on a basic Single-Input Single-Output (SISO) antenna configuration. The BS and each user shall be able to use the whole system bandwidth B_{sys} instantaneously for transmission or reception.

The system bandwidth B_{sys} is divided into a set of orthogonal subchannels $\mathcal{R}_{\text{sys}} = \{1, 2, \ldots, R_{\text{sys}}\}$ of equal width B_{SC} such that $R_{\text{sys}} = B_{\text{sys}}/B_{\text{SC}}$ with $R_{\text{sys}} \in \mathbb{N}$ and $B_{\text{sys}}, B_{\text{SC}} \in \mathbb{R}$. Each of these subchannels represents an allocatable resource and can be used for communication between one user and the BS for the duration of a time slot of length $T_{\text{slot}} \in \mathbb{R}$.

The resulting resource grid is illustrated in Fig. 1. In order to achieve deterministic low latency and to reduce complexity, we constrain the allocation to the frequency domain (i.e.,

This work was supported by the German Federal Ministry of Education and Research (BMBF) as part of the project "USWA - Ultra Scalable Wireless Access" under grant 16KISK256.

subchannels). This implies that all users are served within each time slot; there is no multiplexing of users over time.

B. Resource Allocation

The set of subchannels allocated to user u at time t is denoted as $\mathcal{R}_{\text{user}}(u,t)$ with $u \in \mathcal{U}$ and $t = k \cdot T_{\text{slot}}, k \in \mathbb{N}_0$. An allocation consists of exclusively mapping one or more subchannels to every user, i.e., $\mathcal{R}_{\text{user}}(u,t) \subset \mathcal{R}_{\text{sys}}\ \forall\, u, t$. For now, we assume that every user is allocated the same number of subchannels $R_{\text{user}} \leq R_{\text{sys}}/U$. Note that a user's set of subchannels can be non-contiguous in frequency.

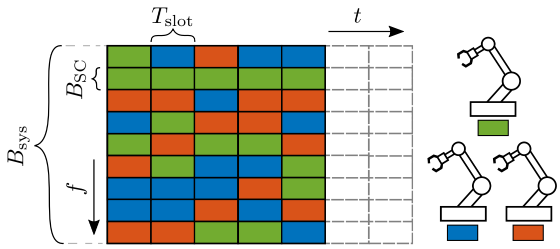


Fig. 1: We multiplex users only in frequency. The allocation problem within a time slot consists of exclusively mapping one or more subchannels to every user.

We assume Channel State Information (CSI) in terms of Signal-to-Noise Ratio (SNR) of each subchannel at every user to be known to the BS, where $\Gamma(r,u,t)$ denotes the instantaneous SNR of the r-th subchannel as observed by user u at time t. This CSI is then used to allocate subchannels to be used at time $t + \Delta t_{\text{alloc}}$. We call Δt_{alloc} the allocation delay.

C. Discrete Outage Definition

Link-level simulations of wireless transmissions between fixed endpoints commonly yield an empirical outage probability (average packet error rate) for a certain channel model and mean SNR. However, to facilitate system-level simulations, we discretize outage events and consider an SNR threshold Γ_{min} under which the receiver is assumed to be in outage (similar to the notion of sensitivity). Γ_{min} then refers to the sum of SNRs of the subchannels allocated to a user: we declare user u to have an individual outage if $\sum_r \Gamma(r,u,t) < \Gamma_{\text{min}}$. If there are one or more individual outages within a given time slot, we declare this a system outage, to which we will refer to simply as outage. Given this definition, we evaluate the reliability of allocators in terms of empirical outage probability (the share of time slots with an outage) and outage duration (product of the number of consecutive time slots in outage and the duration of a time slot).

D. Measurement-Based Channel Emulation

If the system bandwidth B_{sys} is much smaller than the carrier frequency, it is reasonable to assume that a user experiences the same path loss (average SNR) on all subchannels at one instance in time. Differences in the instantaneous SNRs per subchannel originate from multipath propagation conditions, which in turn vary over time depending on the movement of users. It is essential to reproduce temporal correlations in the channel state as we want to determine the duration of outages. Likewise, any investigation of the required granularity of the resource grid (i.e., the subchannel width B_{SC}) requires the channel model to reproduce correlations in frequency. As a user moves through the propagation environment, the SNR distributions per subchannel as well as their correlations in time and frequency may change, e.g., depending on varying Line of Sight (LOS) conditions. Furthermore, the average SNR over the system bandwidth B_{sys} changes as a whole depending on the path loss and hence the distance between a user and the BS. In order to obtain simulation results that are relevant to real-world applications, we need a channel model that reproduces all of these effects for every user (location) and subchannel. Instead of turning to a complicated model with a multitude of parameters that are difficult to choose and justify, we decide to make use of channel data from a comprehensive measurement campaign in a factory hall.

The measurements in [6] were performed in the industrial campus network band at $3.7\,\text{GHz}$ to $3.8\,\text{GHz}$ using an Automated Guided Vehicle (AGV). Channel impulse responses of 512 samples length were captured along a fixed track at an interval of $1\,\text{ms}$ over a duration of $20\,\text{s}$. As the measurements start at a predefined location and the speed of the AGV is known to be $1\,\text{m}\,\text{s}^{-1}$, each point in time can be mapped to a location on the track. By computing the discrete power spectrum for each impulse response, we obtain location-dependent frequency responses. As we are interested in discrete outage events based on threshold comparison, we keep only the magnitude and discard the phase. The measurements' frequency resolution (given by the number of samples (bins) per channel response) determines the smallest resolvable subchannel width, i.e., $B_{\text{SC,min}} = 100\,\text{MHz}/512 \approx 0.2\,\text{MHz}$. Larger subchannels are formed by grouping several bins: the width is an integer multiple of $B_{\text{SC,min}}$ and the combined power equals the sum of the individual powers. Since the SNR is proportional to the measured receive power assuming constant noise power for all subchannels, it is not necessary to estimate the actual SNR because channel-aware allocations only depend on differences in SNR. We can therefore use absolute receive powers in place of SNRs, and we will use outage thresholds based on receive power interchangeably with the SNR threshold Γ_{min}.

The Direct Current (DC) bin and the outermost bins are attenuated for hardware reasons and need to be removed. Since the investigation of different subchannel widths will require a bin count that is integer-divisible, 56 bins at the low end, the DC bin and 55 bins at the high end are discarded in total which yields 400 bins corresponding to $B_{\text{sys}} \approx 78.1\,\text{MHz}$ of usable system bandwidth. Furthermore, we discard the first second of the data which is when the AGV was accelerating, leaving us with $19\,\text{s}$ captured at constant speed.

The channel between a user and the BS, as well as its progression over time, is then emulated by playing back the measured channel frequency responses. Users at different locations can be modeled by using channel data from different points in time, as we describe in the following.

E. Multi-User Scenario with Mobility

We consider a multi-user scenario comprising U closely spaced AGVs following the same track as the AGV in the measurements. This scenario is illustrated in Fig. 2. The experiments in [6] have shown that repeated measurements along the track are highly correlated, which means that channel realizations can be assumed to be constant per location as long as the environment does not change. Thus, we can use the same data set to emulate the channel states of all AGVs. For convenience, we run the simulation at the same time interval as the measurements such that each channel response in the data corresponds to a user state in the simulation.

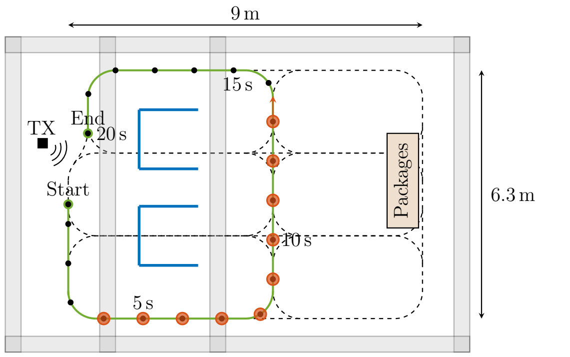


Fig. 2: Map of the original measurement track in a factory hall [6], overlaid with a row of simulated mobile users (red dots). Each position corresponds to a time instance in the measured channel data. Metallic walls in the center (blue lines) cause varying LOS conditions.

The AGVs shall move with a common speed of $1\,\mathrm{m\,s^{-1}}$, which makes a millisecond time offset in the data equivalent to a millimeter track offset in the simulation. Maintaining a common distance of $0.9\,\mathrm{m}$ between each other, the AGVs' channel realizations are spaced by $900\,\mathrm{ms}$ in the data. At the beginning of the simulation, the first AGV is placed at the start of the track which corresponds to the first millisecond of data. The second AGV is placed $0.9\,\mathrm{m} = 900\,\mathrm{ms}$ in front of the first, and so on. In each subsequent millisecond of the simulation, every AGV moves $1\,\mathrm{mm}$ further and its respective channel state corresponds to the next millisecond in the data. When an AGV reaches the end of the track, it wraps around to the beginning. We set the duration of the simulation equal to the duration of the original measurement round such that each AGV passes all possible channel states in the data while traveling a complete round. Even though individual channel states appear multiple times in the simulation (because a given location on the track is passed by all AGVs), this is not a limitation on the system level: the combination of concurrent channel states varies throughout the simulation, which means that the allocator under test encounters different system states anyway and the optimal allocation based on each user's channel state can still be different in each time step.

F. Resource Allocation & Evaluation

We choose the allocation period to equal the simulation and data interval, i.e., $T_{\text{slot}} = 1\,\mathrm{ms}$. Assuming that the users (the AGVs) make use of each allocation period to exchange data with the BS, this communication cycle time agrees well with typical values of closed-loop control applications [7]. Putting together the introduced model and scenario, each time step in the simulation consists of the following tasks:

1) Calculate each user's position along the track based on the current simulation time (Sec. II-E)
2) Retrieve channel state of each user depending on their position (Sec. II-E)
3) Allocate subchannels to users according to a yet to be specified algorithm (Sec. III)
4) Determine and save outage state of each user (Sec. II-C)

After the targeted simulation duration has elapsed (Sec. II-E), the empirical outage probability and maximum outage duration of the system are computed (Sec. II-C).

III. Allocation Algorithms

Finding an analytical solution to the max-min criterion is difficult because the max and min operations are non-linear. On the other hand, finding the optimum by exhaustive search, i.e., by evaluating all possible allocations, quickly becomes infeasible either because of computational complexity.

A. Complexity of the Allocation Problem

Mapping R_{user} subchannels to U users given a pool of R_{sys} subchannels corresponds to drawing R_{sys} for the first user, another R_{sys} for the second user, etc., without put back. The number of possibilities is given by

$$\prod_{i=0}^{U-1} \binom{R_{\text{sys}} - i}{R_{\text{user}}} = \prod_{i=0}^{U-1} \frac{(R_{\text{sys}} - i)!}{R_{\text{user}}!\,(R_{\text{sys}} - i - R_{\text{user}})!} \quad . \tag{1}$$

The factorial in the nominator indicates that the cost of exhaustive search explodes for reasonable values of R_{sys}. And indeed, for a seemingly innocent setup of $R_{\text{sys}} = 10$, $U = 10$, and $R_{\text{user}} = R_{\text{sys}}/U = 1$, there are $3.6 \cdot 10^6$ possible allocations already. For $R_{\text{sys}} = 400$ and $U = 10$, there are over 10^{550} possibilities. There is hence a need for an algorithm that comes to the optimal allocation but requires much less computation than exhaustive search. In the following, we describe the reasoning behind designing a channel-aware low-complexity allocation algorithm and propose one that is later evaluated for different configurations.

B. Algorithm Based on Preference Lists

Recall that the optimal allocation according to the max-min criterion shall maximize the SNR of the weakest user in the system (Sec. II-B). However, the weakest user with respect to the current time slot is identified by the SNR of their associated subchannels, which in turn depends on the allocation. This circular dependency suggests to employ an iterative algorithm where subchannels are allocated one by one: in each iteration, the weakest user is determined anew and gets their preferred subchannel in terms of SNR allocated. Since a user's set of allocated subchannels is only built as the algorithm progresses, we need a metric to identify the weakest user per iteration that is not based on said allocation.

This document is a preprint of: N. Schwarzenberg, A. Trassl, F. Burmeister et al., “Channel-Aware Multi-User Resource Allocation for Ultra-Reliable Low-Latency Communications,” in Proceedings of IEEE International Symposium On Personal, Indoor And Mobile Radio Communications (PIMRC 2023), Toronto, Canada, Sep 2023.

Given the current SNRs $\Gamma(r,u,t)$ for all subchannels $r \in \mathcal{R}_{\text{sys}}$ of every user $u \in \mathcal{U}$, we may keep track of a user's preferred subchannels by sorting them descending by SNR, as illustrated in the first two steps on the left in Fig. 3. The resulting list is called the user's preference list [8] and is denoted as $\mathcal{P}_{u,t,i}$ where i is the current iteration in the algorithm with $i = 0, 1, \ldots, (R_{\text{user}}\, U - 1)$. Each user's list represents a reordered subset of $\mathcal{R}_{\text{sys}}$. It starts with the same number of elements as $\mathcal{R}_{\text{sys}}$ at $i = 0$ and gets smaller in each iteration as subchannels are allocated. A list's elements are hence indexed by $p = 1, 2, \ldots, R_{\text{sys}} - i$. We denote the SNR seen by user u at time t on the subchannel at position p in such list as $\Gamma_{\mathcal{P}}(p)$. The R_{user} top entries of a user's preference list contain the best-case allocation for that user (ignoring other users) and therefore provide a good hint on the user's outcome after allocation. Thus, we propose to calculate a quality Q per user based on the SNRs of some of the subchannels in the current preference list.

Finding the optimal function to derive Q is out of scope for this work; we will instead discuss three reasonable options to get an idea of the problem. A very simple approach is to only look at the first entry in the list (the user's preferred subchannel), which is expressed as $Q_{\text{single}} = \Gamma_{\mathcal{P}}(1)$. However, as subchannel SNRs are subject to fading with decreasing correlation the further they are separated in frequency, a single SNR value is not representative of the user's overall conditions. It seems reasonable to look at a number of $L < R_{\text{sys}} - i$ entries at the top of the list where we call L the allocation lookahead. We suggest $Q_{\text{window}} = \sum_p^L \Gamma_{\mathcal{P}}(p)$ as a metric to incorporate L entries with equal weight. To emphasize steeper gradients (that is, if subchannels further down the list have a much lower SNR), we also propose $Q_{\text{gradient}} = \Gamma_{\mathcal{P}}(1) \cdot \Gamma_{\mathcal{P}}(L)$, just because a product is more sensitive to value differences than a sum.

The user with the lowest Q is considered the weakest and gets its preferred subchannel (top entry in the list) allocated. This subchannel index is then removed from every user's preference list to prevent allocating the same resource to more than one user, and the next iteration begins with calculating new Q for each user. If a preference list is shorter than the lookahead L, the latter is truncated to the length of the list. If a user has been allocated R_{user} subchannels, the user is considered complete and excluded from subsequent iterations. The algorithm is finished when all users are complete. A flowchart of the whole algorithm is depicted in Fig. 3.

Note that this algorithm is guaranteed to end after exactly $R_{\text{user}}\, U$ iterations. Compared to exhaustive search, this is 5 orders of magnitude smaller for $R_{\text{user}} = 1$ and $U = 10$, and 548 orders for $R_{\text{user}} = 40$ and $U = 10$.

C. Baseline Algorithms

In order to assess the outage performance of the above iterative algorithm based on preference lists, we also consider a selection of basic non-iterative algorithms for comparison.

As outlined before, finding the optimal allocation in the sense of max-min by exhaustive search is infeasible. Instead,

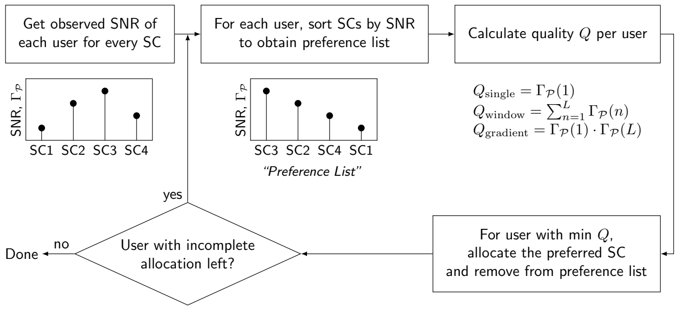


Fig. 3: Simple algorithm for allocating subchannels to users based on preference lists. The algorithm is run every time slot. SNRs per subchannel and user are assumed to be known. Three quality metrics Q computed from a preference list are considered.

we consider an empirical lower bound on outage rate and outage duration by ignoring competition and always allocating the top R_{user} subchannels in a user's preference list to that user. This implies that a subchannel can be allocated to multiple users, but concurrent transmissions on the same subchannel cause unwanted interference and are likely to cause outages, which is why we call this allocator *BestImpossible*.

The above algorithms rely on SNR knowledge about every subchannel at every user, which incurs overhead in a practical system due to additional channel sounding or signaling. Consequently, we are interested in the achievable performance without channel awareness. Since the users in our scenario move at a speed that is low w.r.t. the carrier frequency (see Sec. II-D), SNR values of consecutive time slots are strongly correlated. This is especially relevant if a subchannel experiences an SNR below the outage threshold Γ_{min} because it is likely that the SNR remains below that threshold in the following time slot. Hence, switching subchannels will reduce the duration of outages compared to a static allocation, even without SNR knowledge of the other subchannels. Hopping to different frequencies in a pseudo-random fashion is known from Bluetooth, for example, where it is used to improve reliability as well. We implement this strategy by taking the set of all subchannels $\mathcal{R}_{\text{sys}}$, shuffling it to get a random permutation, and allocating R_{user} entries from that permutation to each user. This ensures that each user's allocation is a random draw from the available subchannels while no subchannel is allocated more than once. We refer to this algorithm as *RandomHopping*.

Finally, we want to compare these dynamic allocators to static allocations that do not change over time. *StaticConsecutive* allocates subchannels consecutively without rearrangement starting with the first subchannel for the first user, such that the allocation for user u at time t is given by $\mathcal{R}_{\text{user}}(u,t) = \{R_{\text{user}}\,(u-1), R_{\text{user}}\,(u-1)+1, R_{\text{user}}\,(u-1)+2, \ldots, R_{\text{user}}\,u\}$. This corresponds to the traditional way of user multiplexing in frequency where a user's bandwidth B_{user} is not split. By contrast, *StaticInterleaved* interleaves the users' bandwidth by assigning the subchannels $\mathcal{R}_{\text{user}}(u,t) = \{u, u+U, u+2U, \ldots, u+(R_{\text{user}}-1)\,U\}$. This exploits more diversity than the consecutive strategy by distributing and hence decorrelating each user's subchannels in frequency.

This document is a preprint of: N. Schwarzenberg, A. Trassl, F. Burmeister et al., “Channel-Aware Multi-User Resource Allocation for Ultra-Reliable Low-Latency Communications,” in Proceedings of IEEE International Symposium On Personal, Indoor And Mobile Radio Communications (PIMRC 2023), Toronto, Canada, Sep 2023.

TABLE I: Common simulation parameters

Carrier Frequency	f_c	$3.75\,\mathrm{GHz}$
System Bandwidth	B_{sys}	$78.1\,\mathrm{MHz}$
Subchannel Width	B_{SC}	$0.195\,\mathrm{MHz}$
Total Subchannels	R_{sys}	400
Number of Users	U	20
Subchannels per User	R_{user}	20
Bandwidth per User	B_{user}	$3.91\,\mathrm{MHz}$

IV. Results

We start the analysis with the smallest possible subchannel width, see Sec. II-D. While this configuration incurs the maximum number of subchannels in the system and hence the highest algorithmic complexity, it also offers the most degrees of freedom to optimize the allocation and maximize reliability. The maximum realistic user count is determined by the physical scenario described in Sec. II-E. Assuming a quite close but physically feasible spacing of $0.9\,\mathrm{m}$ between AGVs, we can fit 21 users on this track, which we round to 20 to get equal integer subchannel counts per user. We focus on the case of full system load, which means that all subchannels in the system will be allocated to users. This represents the most challenging allocation scenario as users are more likely to compete for the same subchannels, and offers the highest spectral efficiency. A summary of common simulation parameters is given in Tab. I.

As explained in Sec. II-D, receive power and SNR are interchangeable when it comes to allocation. In the following, we provide readings of an outage threshold in terms of power spectral density instead of the previously defined SNR threshold Γ_{min}. The given values reflect the actual powers from the factory measurement campaign and are normalized to the user bandwidth B_{user} for better comparability. Note that absolute values of power spectral density (or SNR) are not relevant to the results and will only be used when referring to certain parts in the plots. A lower outage threshold can be interpreted as less noise and hence better channel conditions, or equivalently, as a receiver configuration (e.g., in terms of modulation and coding) with lower sensitivity level.

A. Comparison of Allocators

We begin by comparing the allocators introduced in Sec. III by their achieved reliability in terms of maximum outage rate and maximum outage duration. The maximum outage rate corresponds to the average outage rate of the weakest user; the maximum outage duration is the longest outage event regardless of the user. By regarding only the maxima, we assess an allocator by the worst case performance it causes for any user in the simulation. This complements the connected robotics use case described in Sec. I where a failure of any device causes a system failure and needs to be avoided.

The plots in Fig. 4 quickly show that the way of allocating subchannels to users has significant impact on the system's reliability. The curve of *StaticConsecutive* marks the baseline performance of static user multiplexing in frequency. By distributing each user's bandwidth over the system bandwidth (*StaticInterleaved*), we leverage frequency diversity: here, a user's subchannels are spaced by $B_{\text{SC}}\,U \approx 4\,\mathrm{MHz}$ which is approximately equal to the 90 % coherence bandwidth of this scenario [6]. The diversity gain becomes visible as steeper slope in the curve and yields a much reduced outage rate by more than two orders of magnitude for $-140\,\mathrm{dBm/Hz}$. The third allocation scheme without the need for channel knowledge is *RandomHopping*, which offers a similar performance as *StaticInterleaved*. *RandomHopping* performs slightly worse however, because on average, randomly allocated subchannels can be closer to each other than with *StaticInterleaved*. On the other hand, looking at the duration of outages, *RandomHopping* performs better than *StaticInterleaved* because an allocation time slot is short compared to the channel's correlation time due to the low speed of $1\,\mathrm{m\,s^{-1}}$ and the moderate carrier frequency of $3.75\,\mathrm{GHz}$. Thus, changing subchannels between time slots reduces the chance of consecutive outages and therefore the maximum outage duration.

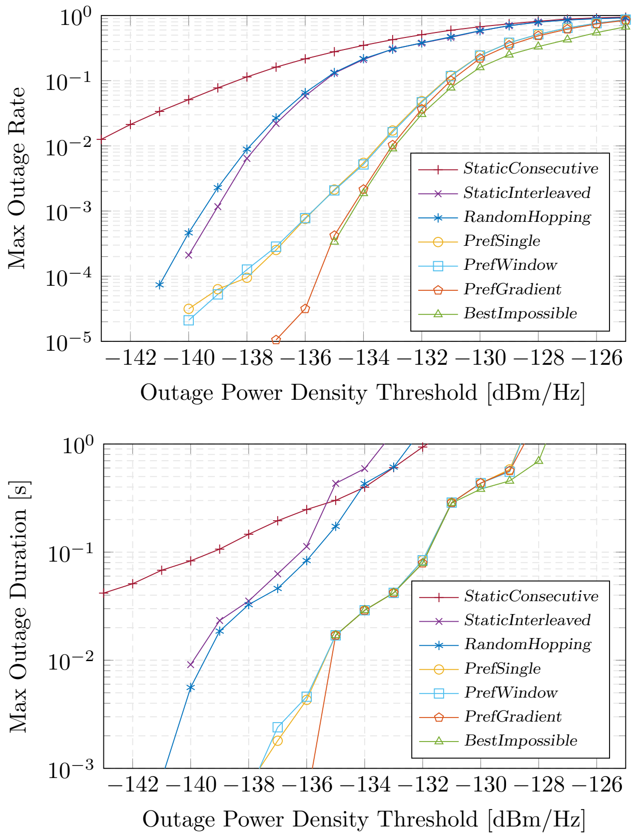


Fig. 4: Comparison of allocators by maximum outage rate (top) and maximum outage duration (bottom) versus outage power density threshold.

As expected, all of the considered channel-aware allocators perform much better in terms of outage rate and outage duration. For example, at an outage rate in the range of 10^{-3} to 10^{-2}, channel-aware allocation tolerates a $4\,\mathrm{dB}$ to $6\,\mathrm{dB}$ higher outage threshold than *StaticInterleaved* and *RandomHopping*.

Down to an outage rate of about 10^{-3}, all of the allocators based on preference lists are fairly close to the (unfeasible) lower bound marked by *BestImpossible*. For a lower outage rate and threshold, we see that *PrefSingle* and *PrefWindow* run into an error floor (outage rate decreases less for decreasing threshold) at a threshold of about $-138\,\mathrm{dBm}$ while *PrefGradient* stays closer to *BestImpossible*. We conclude that the quality metric Q for assessing which user is the weakest during allocation (Sec. III-B) becomes more important as we get into the very low range of outage rate and duration that is typical for URLLC use cases. This can also be understood in the sense that suboptimal allocations have a higher impact when users depend on all of their allocated subchannels to reach the required threshold. Interestingly, *PrefWindow* doesn't perform much better than *PrefSingle* even though it uses the same range of subchannels (lookahead L) as *PrefGradient*. Moreover, *PrefWindow* incorporates SNRs from all subchannels in this range while *PrefGradient* only looks at the first and L-th subchannel. This suggests that subchannels further down the preference list become more important as we reach a low outage rate. We conclude that allocations may improve if users are assessed using more than the top subchannel of their preference list. Finding the optimal function for the quality metric Q is left for further research.

B. Impact of Subchannel Width

To keep the complexity as low as possible, we are interested in the trade-off between increased subchannel width and the potential loss of reliability due to less fine-grained allocation. Subchannels wider than a single measured frequency bin (Sec. II-D) are modeled by grouping several bins together which results in a grid of fewer resources. A new subchannel's SNR is then given by the sum of the grouped bins' SNRs. Possible subchannel widths have to be integer multiples of $B_{\text{SC,min}}$ to avoid the need for interpolation between bins. For instance, we can divide the system bandwidth into subchannels of $B_{\text{SC}} = 4\,B_{\text{SC,min}} \approx 0.8\,\mathrm{MHz}$, resulting in $R_{\text{sys}} = 400/4 = 100$ and $R_{\text{user}} = 40/4 = 10$. Since R_{user} also has to be integer, the set of possible subchannel widths is limited. Note that the following investigation is done for a reduced number of users $U = 5$ to be able to evaluate larger subchannel widths.

A performance comparison of the baseline allocator *RandomHopping* as well as the channel-aware allocator *PrefGradient* is shown in Fig. 5. Each subplot shows outage rate and duration, respectively, over subchannel width. The curves in each subplot correspond to different outage power density thresholds Γ_{min}. For all allocators, the maximum outage rate starts to slowly deteriorate at about $0.8\,\mathrm{MHz}$ and rises quickly from about $4\,\mathrm{MHz}$ and above. Not surprisingly, these values are in the order of magnitude of the coherence bandwidth in this scenario [6]. While this trend is visible as well for the maximum outage duration of *PrefGradient*, the reverse is true for *RandomHopping*: larger subchannels average out narrow deep fades and the risk of hitting bad resources consecutively is actually reduced. It can further be seen that the performance impact is bigger for lower outage rates and durations. In other words, low performance applications can get away with coarse allocation and reduced complexity, but for achieving the highest reliability, the allowable subchannel width depends on the environment's coherence bandwidth. For this particular scenario, without loss of reliability, a subchannel width of $0.8\,\mathrm{MHz}$ reduces the number of system resources by a factor of four as compared to the initial width of about $0.2\,\mathrm{MHz}$.

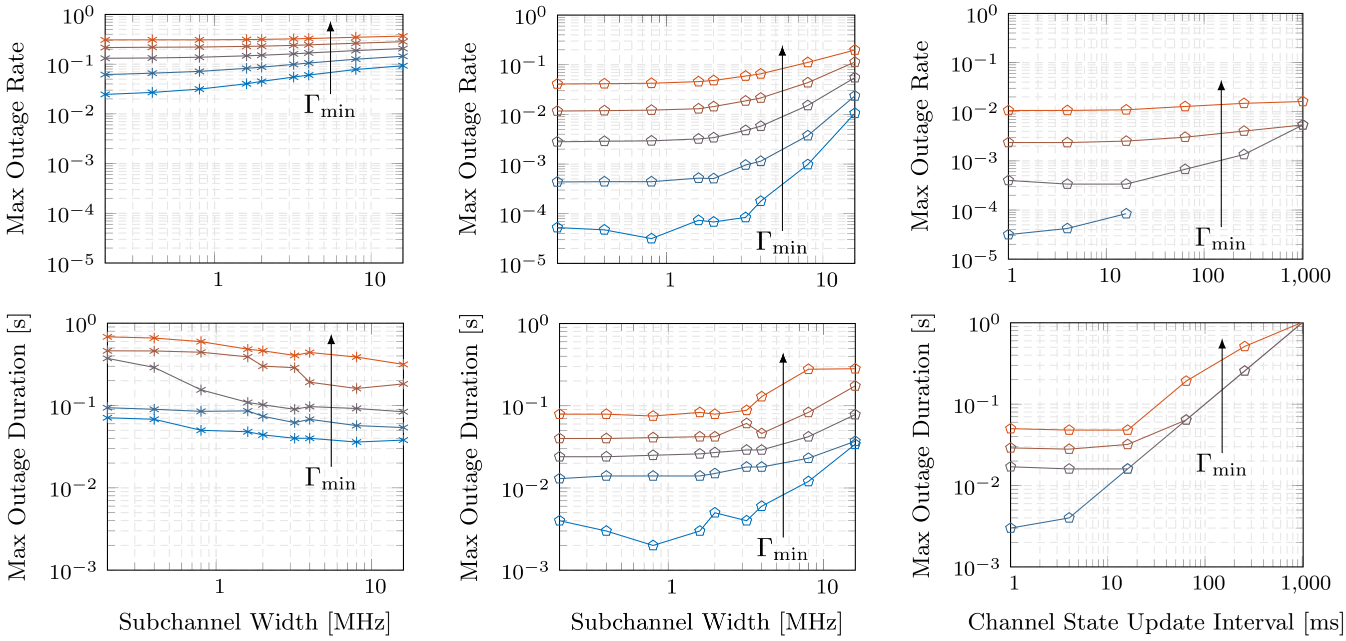
C. Impact of Allocation Delay

So far, the channel-aware allocators had up-to-date CSI in terms of SNR at every system subchannel and for each user available. This means that the time between CSI acquisition and allocation, in the following referred to as allocation delay Δt_{alloc} (Sec. II-B), is zero. However, in a real-world system, CSI needs to be obtained first by collecting channel estimates. Since communication between a single user and the BS only uses some of the system subchannels at a time, information about the other subchannels probably requires additional airtime to transmit sounding reference signals. While the design of an efficient scheme to obtain CSI at the BS is out of scope of this work, such design would benefit from knowing the required CSI update interval $\Delta t_{\text{CSIupdate}}$ to avoid sacrificing reliability due to bad subchannel allocation based on outdated information.

To this end, the simulation is adapted such that SNR data per subchannel and user are kept constant for the duration of $\Delta t_{\text{CSIupdate}}$ before being updated to the current data. This implies a linearly growing allocation delay $\Delta t_{\text{alloc}} \in [0, \Delta t_{\text{CSIupdate}})$ that is reset to zero after $\Delta t_{\text{CSIupdate}}$.

The results for allocator *PrefGradient* are shown in Fig. 5c. Similar to the impact of increased subchannel width, bigger update intervals have greater impact on low outage rates, that is, for high reliability applications. At an outage rate of about $3 \cdot 10^{-4}$, performance does not suffer for an update interval up to $10\,\mathrm{ms}$, which is in the order of the coherence time of this environment. Whereas for $3 \cdot 10^{-5}$, already $4\,\mathrm{ms}$ incur a noticeable loss of reliability. Regarding the duration of outages, the turning points in terms of update interval are the same, but operating points at higher values (lower reliability) are affected as well. This is because the allocation is not changed in time while the currently selected subchannels are fading. Of course, a subchannel may also improve over time and thereby compensate others that fade. This is why for an averaged metric like the outage rate, the impact is less visible (compare with the performance of the *Static** allocators in Sec. IV-A). However, the duration plots show only the maxima of the outage durations in a simulation which puts rare events of longer outages into focus.

Channel-unaware allocators such as *RandomHopping* do not rely on CSI and are therefore not affected by increased update intervals. In cases when CSI cannot be kept up to date, it might be tempting to fall back to such an allocator to keep outage durations low. Comparing the results in Fig. 5c to Fig. 4 for an outage power density threshold of $-133\,\mathrm{dBm/Hz}$ (top curve in Fig. 5c), *RandomHopping* is at a maximum outage duration of $0.6\,\mathrm{s}$. This value is only undercut by *PrefGradient*



(a) Allocator: *RandomHopping* (b) Allocator: *PrefGradient* (c) Allocator: *PrefGradient*

Fig. 5: Maximum outage rate and maximum outage duration over subchannel width (a–b) and CSI update interval (c) for outage power density thresholds -137 to $-133\,\mathrm{dBm/Hz}$ (curves from bottom to top), compared for different allocators at $U = 5$.

for update intervals greater than $400\,\mathrm{ms}$. However, at that point, *RandomHopping* is already at a high outage rate of $3 \cdot 10^{-1}$, which is not a relevant operating point of URLLC.

V. Conclusion

This work shows that adaptive allocation of non-contiguous resources improves reliability by orders of magnitude compared to static and contiguous allocations. Finding the optimal collision-free allocation by exhaustive search becomes computationally unfeasible for hundreds of subchannels and tens of users. In lieu of the optimal allocation, an optimistic best-case performance bound is given by the allocation that grants each user their preferred subchannels without taking into account collisions. A low-complexity channel-aware allocator based on preference lists is proposed and found to be near-optimal. Outage rates and especially outage durations are much improved over channel-unaware allocators because weak subchannels at a given user's position are effectively avoided. Using the *Gradient* quality function, the algorithm's performance gets within a fraction of a dB from the best-case performance bound. Regarding the required granularity of the resource grid, the subchannel width should be kept well below the expected coherence bandwidth and the CSI update interval below the coherence time. Since the coherence time is a function of velocity, different update intervals per user could be considered depending on the user's maximum velocity. In the event that the CSI update interval and subchannel width are larger than indicated by the coherence metrics, the channel-aware preference list allocator still outperforms the channel-unaware schemes in the low outage rate regime. As the channel-aware allocators rely on up-to-date CSI for each user, future work might investigate suitable schemes for multi-user channel sounding. Further open topics include the evaluation and fine-tuning of the allocators for different coherence bandwidths, the optimization of the proposed algorithm towards lower complexity to ensure that real-time allocation does not become a latency bottleneck, and a quantitative reliability comparison of max-min resource allocation to state-of-the-art throughput-maximizing allocation.

References

[1] J. Rao and S. Vrzic, “Packet Duplication for URLLC in 5G: Architectural Enhancements and Performance Analysis,” *IEEE Network*, 2018.

[2] A. Lopez-Raventos and B. Bellalta, “Multi-Link Operation in IEEE 802.11be WLANs,” *IEEE Wireless Communications*, 2022.

[3] L. Massoulie and J. Roberts, “Bandwidth sharing: objectives and algorithms,” *IEEE/ACM Transactions on Networking*, 2002.

[4] Z. Han, Z. Ji, and K. R. Liu, “Fair multiuser channel allocation for OFDMA networks using Nash bargaining solutions and coalitions,” *IEEE Transactions on Communications*, 2005.

[5] A. Traßl, P. Schulz, L. Scheuvens, N. Schwarzenberg, and G. Fettweis, “On the Outage Probability of Channel Prediction Enabled Max-Min Radio Resource Allocation,” in *IEEE Wireless Communications and Networking Conference (WCNC)*, Austin, USA, Apr 2022.

[6] F. Burmeister, N. Schwarzenberg, T. Hößler, and G. Fettweis, “Measuring Time-Varying Industrial Radio Channels for D2D Communications on AGVs,” in *IEEE Wireless Communications and Networking Conference (WCNC)*, Nanjing, China, Mar 2021.

[7] M. Gundall, J. Schneider, H. Schotten, M. Aleksy, D. Schulz, N. Franchi, N. Schwarzenberg, C. Markwart, R. Halfmann, P. Rost, D. Wübben, A. Neumann, M. Düngen, T. Neugebauer, R. Blunk, M. Kus, and J. Grießbach, “5G as Enabler for Industrie 4.0 Use Cases: Challenges and Concepts,” in *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Turin, Italy, Sep 2018.

[8] T. Hößler, M. Simsek, and G. Fettweis, “Matching-Based Resource Allocation for Multi-User URLLC in Unlicensed Frequency Bands,” in *International Symposium on Wireless Communication Systems (ISWCS)*, Oulu, Finland, Aug 2019.