System Analysis of State-Aware Resource Allocation for Closed-Loop Control Systems

Lucas Scheuvens[1], Philipp Schulz[1], Tom Hößler[2], Norman Franchi[1], André Barreto[2], and Gerhard P. Fettweis[1,2]
[1]Vodafone Chair Mobile Communications Systems, Technische Universität Dresden, Germany
[2]Barkhausen Institut, Dresden, Germany
Email: {lucas.scheuvens, philipp.schulz, norman.franchi, fettweis}@ifn.et.tu-dresden.de,
{tom.hoessler, andre.nollbarreto}@barkhauseninstitut.org

***Abstract*—Wireless closed-loop control is of major significance for different application areas, such as future industrial manufacturing, and ultra-reliable low-latency communications (URLLC) are designed to enable such systems. Static multi-connectivity, in which a number of independent parallel channels are allocated for each service, is a possible solution to achieve URLLC requirements, but this increases resource usage significantly, which becomes an issue particularly in multi-user systems. Building upon a control-communications codesign (CoCoCo) approach, a control-application optimized state-aware resource allocation (SARA) scheme was developed, which exploits the control cycle's inherent capability of tolerating a limited number of consecutive packet losses before ultimately failing. In essence, SARA negatively correlates packet losses through dynamic channel allocation in order to yield extraordinary availability values while keeping the average resource consumption low. This article develops a multi-user system representation of SARA with competition for limited resources using a Markov chain approach and subsequently evaluates the mean time to failure, demonstrating that SARA scales better than static multi-connectivity, fully supporting the maximum system availability at fewer channels per agent.**

***Index Terms*—Control Communications Co-Design, Wireless Control, Adaptive Resource Management, System Analysis, State-Aware Resource Allocation, Multi-Connectivity**

I. Introduction

Enabling closed-loop control systems over wireless communications channels is of major significance for future manufacturing, since it will enable a hitherto unknown degree of flexibility and productivity [1]. Currently in peer-reviewed literature, there exist two main approaches in order to achieve this goal: On one hand, in the communications research community, ultra reliable low latency communications (URLLC) - as one of the main 5G pillars - is still considered to be the enabling technology, as it is designed to achieve deterministic latency values lower than $1\,\mathrm{ms}$ and reliability values larger than $99.9999\,\%$, the higher objective being to replace and extend widely established wired industrial communications systems [2]. In this context, 3GPP defines *reliability* as the complement of the packet loss rate (PLR), i.e., $1 - \mathrm{PLR}$, whereby a packet is considered lost when it is not transmitted within the time constraint required by the targeted service [3]. Achieving a PLR lower than 10^{-6}, however, while simultaneously achieving a deterministic latency lower than $1\,\mathrm{ms}$ and one-way payload data rates exceeding $90\,\mathrm{Mbit/s}$, as for instance with Ethercat [4], will require an immense amount of wireless resources, due to the need for high-bandwidth, high-order multi-connectivity (MC), as well as little to no tolerance for retransmissions (due to the low-latency constraint) [5], [6]. On the other hand, the control engineering research community has been studying so-called networked control systems (NCS) since the late 1990s. The main motivation for NCS stems from reduced costs because general-purpose and low-cost communications systems are cheap and universally available. As these systems usually do not guarantee quality of service (QoS), NCS deal with the question of how to cope with communications imperfections on the control side in order to maximize control utility. The imperfections include but are not limited to delay [7], packet drop-out [8], competition for resources [9], and data rate limitations [10].

In this context, Industry 4.0 is an omnipresent buzzword describing the shift towards high-performing all-connected manufacturing. However, it is doubtful that this vision can be put into practice with unmanaged, unoptimized communications systems as they are used in NCS. At the same time, the high resource consumption in URLLC raises the question whether URLLC can also be applied in a resource-limited multi-user system, which is an essential quality to turn Industry 4.0 into reality. Hence, we propose applying appropriate adjustments in both domains, control and communications. In literature, this approach, which we name co-design of control application and (wireless) communications (CoCoCo), is fairly new. It includes the design of wireless communications networks that are able to deal with the trade-off between costly wireless resource utilization and control performance. We propose the development of communications systems that have a well-understood interface with the control application in order to decide in real-time the importance of each transmitted packet. This has the potential to save a vast amount of valuable resources, depending on the control application under consideration.

This article extends our work in [11] that presented a novel, control-application optimized resource allocation approach termed state-aware resource allocation (SARA). Since many control applications can tolerate a few consecutive lost packets, SARA dynamically assigns a certain number of parallel independent channels to agents in order to increase reliability through MC, depending on the number of previously lost packets. Because of this, it might cause issues in a multi-user system, due to the competition for limited resources. Hence, this article targets to answer SARA's applicability potential on a system level based on the performance metrics mean time to failure (MTTF) and channel utilization and is structured as follows. In Section II, an individual agent's failure model is presented, precisely defining the operational bounds of a wireless control application. In Section III, the aforementioned SARA is outlined, providing extremely high availability at low average resource consumption. In Section IV, a system-wide failure model is developed that covers the constraint of a limited number of available channels. Section V demonstrates evaluation results before Section VI concludes the article.

The contributions of this article are

- an extension of an individual failure model to a system model using a Markov chain approach,
- the closed-form key performance indicator (KPI) derivation (system-wide mean time to failure $\mathrm{MTTF}_{\mathrm{sys}}$, channel utilization η) for appropriate assessment of system performance,
- the computational complexity analysis, and
- the verification with system simulations.

II. Agent Failure Model

The term *availability* commonly denotes the probability of successfully transmitting a packet [12]. It can be increased through the use of frequency, time, space, and/or code diversity. For closed-loop control applications however, diversity in time, i.e., retransmissions of the *same* data – which also extends to hybrid repetitions such as Hybrid Automatic Repeat Request (HARQ) – is often not feasible due to the data being outdated fast, deeming the retransmission useless [7], [13]. Hence, in this work, only diversity schemes supporting simultaneous transmissions are considered and we more specifically limit ourselves to frequency diversity for simplicity.

In this article, all channels that are assigned simultaneously to an agent[1] are assumed to have a frequency spacing larger than the coherence bandwidth. The packet interarrival time is assumed longer than the coherence time, which is reasonable for control applications that feature sampling periods $T_{\mathrm{s}} > 10\,\mathrm{ms}$, assuming a carrier frequency $f_{\mathrm{c}} = 3.75\,\mathrm{GHz}$ and a low mobility scenario with the maximum speed $v_{\mathrm{max}} = 2\,\mathrm{m\,s^{-1}}$ [14]. Also, straight-forward decorrelation techniques such as channel hopping may be applied that further alleviate the issue of temporal correlation [15]. In consequence, all transmissions are assumed to be independent in both frequency and time. In addition, we assume that all channels exhibit a fixed per-channel packet loss probability p_{loss}. For a number of simultaneously used channels L, a selection combining (SC) scheme is considered because of its low complexity, which enables the combination of channels in

[1]An agent in this article is one instance of the control application under consideration, e.g, one automated guided vehicle (AGV). However, it is emphasized that this article does not only refer to AGVs but to other control applications as well.

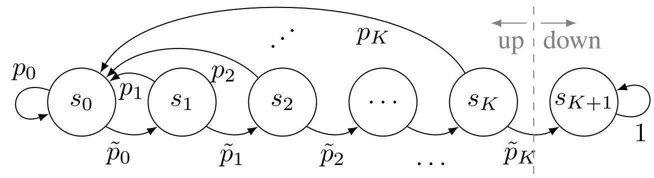

Figure 1. Single-agent state model.

higher network layers. With these assumptions, the availability is given as

$$A_{\mathrm{com}} = 1 - p_{\mathrm{loss}}^{L} \ . \tag{1}$$

A_{com} describes a communications-related metric (as denoted by the subscript), and therefore does not carry information about the availability of the control application, unless every packet transmission failure also triggers a control application failure. As many control applications are inherently able to tolerate packet losses to some extent [9], [14], this is usually not the case. Since in the context of industrial automation *application-related* metrics are imperative, availability from the communications domain needs to be mapped to availability in the control domain.

This conversion requires a fundamental understanding of how packet losses affect the control loop and, more particularly, under which circumstances to shut down the control application due to communications service failure. We have shown in our previous work that the AGV control application is able to tolerate packet losses as long as not *too many* occur *consecutively* [14]. In essence, this spans a *survival time* of the control application when using the terminology of [3]. Let K describe the number of consecutive packet losses that the control application is able to tolerate. Consequently, if $K+1$ packets are lost consecutively, the control application will be shut down in order to avoid damage or, more severely, human harm. The specific value of K differs across the control application under investigation, the required control performance and also the chosen packet interarrival time, which is assumed equal to the sampling period T_{s} of the control application. For the AGV use case cited above, $K = 3$ was determined at a sampling period $T_{\mathrm{s}} = 45\,\mathrm{ms}$.

Following the availability definition in [16], the communications-related mean time to failure $\mathrm{MTTF}_{\mathrm{com}}$ (average time until a transmission fails) can be expressed as

$$\mathrm{MTTF}_{\mathrm{com}} = \frac{T_{\mathrm{s}}}{1 - A_{\mathrm{com}}} \ . \tag{2}$$

This conversion is performed because the mean time to failure is a more tangible quantity of reliability theory and shall be used henceforth.

In order to convert between $\mathrm{MTTF}_{\mathrm{com}}$ and the control-application-related MTTF (without subscript), we introduced in [11] the Markov Model depicted in Fig. 1. Each agent is assigned a state $s_k \ \forall \ k \in \{0, \dots, K+1\}$, where the subscript denotes the number of consecutive packet losses that occured immediately before entering s_k. For instance, in s_2 the last two transmissions failed. Consequently, when a transmission succeeds, the state s_0 is entered. Whenever a packet is lost with transition probability $\tilde{p}_k$, the state index is incremented by one. The rightmost state s_{K+1} denotes the only failure state because in order to enter it, $K+1$ consecutive packet losses need to occur. All other states are considered "up". The MTTF of a single agent when initializing in s_0 was derived to be [11]

$$\mathrm{MTTF} = T_{\mathrm{s}} \boldsymbol{e}_0 \boldsymbol{N} \boldsymbol{1} \tag{3}$$

with $\boldsymbol{e}_0 = [1\ 0\ \dots\ 0]$, $\boldsymbol{1} = [1\ \dots\ 1]^\intercal$ and $\boldsymbol{N} = (\boldsymbol{I} - \boldsymbol{Q})^{-1}$ as the fundamental matrix of the absorbing Markov chain [17]. Therein, $\boldsymbol{Q}$ denotes the transition probability matrix of all transient states.

By changing the number of parallel channels L_k assigned in each state s_k of the Markov model, the transition probabilities $p_k = 1 - \tilde{p}_k = 1 - p_{\text{loss}}^{L_k}$, see (1), can be tuned such that a desired MTTF can be achieved. The next section introduces SARA, which achieves extraordinarily high MTTF values while keeping the average number of channels that are assigned to each agent low.

III. State-Aware Resource Allocation

SARA was designed to increase the MTTF while keeping the average number of channels that are assigned to each agent low. In our previous work [11], we compared static resource allocation with 1, 2, and 3 channels to schemes that assign few resources in low states and increase the amount of parallel channels as k increases. The physical layer (PHY) of each agent is assumed to be able to demodulate up to $L_{\text{a,max}} = 4$ parallel channels.

We adopt the following notation from [11]. The adaptation schemes follow a regular pattern and are denoted as S_l^j, with l indicating the base number of channels, i.e., the number of channels allocated after a successful transmission; and j indicating the number of channels added for each lost packet. $S_{L_{\text{fix}}}^0$ corresponds to a MC approach with L_{fix} fixed channels, termed *static* schemes in the following. Whenever a packet is transmitted successfully, the number of channels is reset to the base value of the scheme. As examples in this article, we consider the schemes S_1^0, S_2^0, S_3^0, S_1^1, and S_2^1.. Please note that for S_2^1, $L_{\text{a,max}} = 4$ is already reached in s_2, hence, the amount of channels cannot increase in s_3. In the rest of this article $K = 3$ is considered, which denotes a conservative value considering the design guidelines of [18] that recommend a 10- to 20-fold oversampling rate, however, the analysis can be performed for any values of l, j, and K, and also for arbitrary schemes that do not follow the regular pattern stated above, e.g., schemes that only react after the second lost packet.

The dynamic resource allocation of SARA that exploits the characteristics of closed-loop control was proven to increase the control application's MTTF by orders of magnitude compared to static MC while simultaneously keeping the average resource consumption low [11] and only sacrificing minor control utility [14]. For instance at $K = 3$, the MTTF improves by 100x from 34 days to 10 years for the SARA scheme S_1^1 compared to static dual-connectivity S_2^0 while only using approximately half the amount of parallel channels on average (1.09 compared to 2). This demonstrates SARA's outstanding potential. However, while for a single agent the potential is high, the dynamic assignment of resources might lead to poor scalability. This is, when the number of available channels is limited, the SARA scheme might drastically reduce the system-wide $\mathrm{MTTF}_{\text{sys}}$ due to the requested SARA channels not being disposable to all agents. The only way to ensure a truly unimpaired system-wide $\mathrm{MTTF}_{\text{sys}}$ is the deployment of $L_{\text{a,max}} M$ channels (with M denoting the number of agents) as this will ensure that all agents will always be assigned the amount of channels they request. However, this is certainly wasteful, especially considering the wide range of up to $L_{\text{a,max}} = 4$ parallel channels for each agent. Reducing the amount of available channels in the whole system to $L_{\text{av}} < L_{\text{a,max}} M$ entails not being able to assign all requested resources sometimes. This might reduce the $\mathrm{MTTF}_{\text{sys}}$ and therefore ultimately lead to a premature failure. Hence, in the next section, a system extension of the model presented in Fig. 1 is developed that incorporates a maximum amount of available channels L_{av} and multiple agents.

IV. Modeling a SARA System

We introduce a system of M agents from which each individually operates according to the Markov chain depicted in Fig. 1. This means that each agent is able to tolerate K consecutive packet losses before ultimately failing, i.e., reaching the absorbing state s_{K+1}. We introduce a new superimposed Markov chain that agglomerates all individual agent states to a single system state

$$S_{|s_0|,|s_1|,\dots,|s_{K+1}|}\ . \tag{4}$$

Thereby, $|s_k|$ denotes the amount of agents that currently reside in state s_k. Consequently, $\sum_{k=0}^{K+1} |s_k| = M$. Please note that for the system state only the amount of agents in each state s_k matters and not the specific set of agents.

We define that the system is *down* when at least one agent is in the down state, i.e., $|s_{K+1}| > 0$, else it is *up*. All *down* states are collapsed to a single down state S_{d} and the last index in (4) corresponding to $|s_{K+1}|$ is dropped from the notation as it will take the value 0 for all *up* states. For more concise notation, we introduce a linear index $i \in \{0, \dots, Z_{\text{up}} - 1\}$ for all system states, with Z_{up} constituting the number of *up* system states.

A. Assumptions

In this article, perfect knowledge about each agent's state is assumed through ideal acknowledgments. Erroneous acknowledgment (ACK) transmissions might lead to wrong state estimates, but signalling information such as ACKs can be protected with low-rate powerful codes and have a PLR orders of magnitude lower than that of data packets. Therefore, for simplicity, we assume that ACKs are always correctly received. We introduce an amount of L_{av} available channels to the system. Each set of up to $L_{\text{a,max}}$ parallel channels can be assigned to any agent. With $L_{\text{req}}(S_i)$ as the total amount of requested channels (by all agents) in state S_i, it follows that the system does not have enough parallel channels if $L_{\text{av}} < L_{\text{req}}(S_i)$. Hence, $L_{\text{req}}(S_i) - L_{\text{av}}$ channels need to be denied by admission control, constituting a system-induced deviation from the ideal SARA resource allocation. In this article, channels will be denied randomly until $L_{\text{req}}(S_i) = L_{\text{av}}$. From an agent's perspective, this implies a weighting by the number of channels each agent is requesting. We stress that whenever an agent was denied a channel, this particular agent is still susceptible to be denied another (if it still has at least one) as long as $L_{\text{req}}(S_i) > L_{\text{av}}$. This also means that there might be agents that are denied all requested channels, subsequently leading to an inevitable packet loss and consequently a transition from $s_k \to s_{k+1}$ for that particular agent.

B. Transition Probabilities

In order to derive the transition probabilities of the system Markov chain, we revisit to the individual agent Markov chain in Fig. 1. The individual state transition probabilities p_k are determined by the chosen resource assignment scheme. The transition probabilities and, consequently, also the system's mean time to failure MTTF_{sys} depend highly upon the number of available channels L_{av}, and the base channel allocation. The approach to derive the transition probabilities is straightforward combinatorics and the details shall be omitted for conciseness. It is summarized by the following steps.

1) Fix M, K, L_{av}, and the resource allocation scheme. Keep in mind that the resource allocation scheme can only be implemented for *every* agent in a particular time step if enough channels are available. Channels may be randomly denied if, in total, too many are requested.
2) Calculate the set of all *up* system states according to (4).
3) For every *up* system state, calculate (a) all possible sink states and (b) all possible channel allocations and their respective probability. Note that there are a multitude of possible channel allocations per system state, each with individual probability.
4) Determine the probability of reaching each possible sink state for each possible channel allocation via (1) and combinatorics.
5) Combine each channel allocation probability with the probability of reaching a given sink state with this particular channel allocation.
6) Combine these probabilities for each sink state.

C. KPI Derivation

The derivation of the MTTF_{sys} is a well-known procedure for absorbing Markov chains and is performed analogously to (3). Hence, when initializing in S_0 (all agents start in s_0),

$$\text{MTTF}_{\text{sys}} = T_{\text{s}} \boldsymbol{e}_0 \boldsymbol{N}_{\text{sys}} \mathbf{1} \tag{5}$$

where $\boldsymbol{N}_{\text{sys}}$ denotes the fundamental matrix of the system Markov chain and can be derived from the system's transition probability matrix. Additionally, let $\mathbf{1}(L_{\text{av}})$ denote a $(Z_{\text{up}} \times 1)$ vector whose elements are composed through

$$\mathbf{1}(L_{\text{av}})_i = \begin{cases} 1 \text{ if } L_{\text{req}}(S_i) \geq L_{\text{av}} \\ 0 \text{ else} \end{cases} . \tag{6}$$

Then, we introduce

$$\eta(L_{\text{av}}) = \frac{\boldsymbol{e}_0 \boldsymbol{N}_{\text{sys}} \mathbf{1}(L_{\text{av}})}{\boldsymbol{e}_0 \boldsymbol{N}_{\text{sys}} \mathbf{1}} \tag{7}$$

that we term *channel utilization*, describing the proportion of time in which all L_{av} channels are in use. It represents how often all the channels are requested, and, hence, indicates the value of adding an additional channel.

D. Computational Complexity

The number of *up* states can be derived through fundamental combinatorics as

$$Z_{\text{up}} = \frac{(M+K)!}{M!K!} \tag{8}$$

and therefore scales with $\sim M^K$ for large M and small K. The number of transitions N_{trans} originating in system state $S_{|s_0|,\ldots,|s_K|}$ can be derived as

$$N_{\text{trans}}(S_{|s_0|,\ldots,|s_K|}) = \begin{cases} \prod_{k=0}^{K} (|s_k|+1) & \text{if } |s_K| = 0 \\ \prod_{k=0}^{K-1} (|s_k|+1) + 1 & \text{otherwise} \end{cases} \tag{9}$$

with the case discrimination stemming from merging all *down* states to S_{d}. The number of transitions $N_{\text{trans,total}}$ in the whole Markov chain therefore scales with $\sim M^{2K}$. In other words, the growth in M is polynomial (exponent K for the number of states and $2K$ for the number of transitions) while in K it is exponential (with base M).

The admission control scheme of this article that denies requested channels randomly if $L_{\text{req}} > L_{\text{av}}$, features a multitude of *micro-transitions* between states and therefore increases the computational complexity significantly. These micro-transitions stem from many possible channel assignments – resulting from the random denying – that lead to the transition from one given source state to one given sink state, resulting in different probabilities that need to be stochastically combined. Due to this complexity and limited computing resources, we limit $M \leq 20$ for this article.

V. Evaluation Results

This section first compares analytical with simulation results for an example. It then proceeds to compare and discuss the analytical results for $M = 20$ and all resource allocation schemes of this article. The focus is especially on the potential of each scheme to be applied in a multi-user system with competition for resources.

For the comparison of the analytical and simulation results, the per-channel packet loss probability is set to $30\,\%$ in order to shorten the simulation duration significantly. For all other proper evaluations, it will be decreased to $10\,\%$,

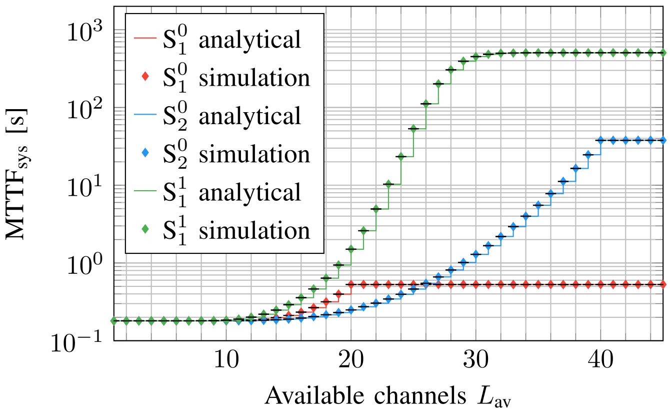

Figure 2. Verification of closed-form $\mathrm{MTTF_{sys}}$ expression through extensive system-level simulation.

following the throughput-optimizing design recommendations in [19], enabling a high spectral efficiency. Also, following our previous work in [14], the sampling period of the control application (equal to the inter-transmission time interval) is set to $T_{\mathrm{s}} = 45\,\mathrm{ms}$. Please note that the maximum system-wide $\mathrm{MTTF_{sys}}$, $\mathrm{MTTF_{sys,max}}$, which is achieved without any limitation of channels, decreases by a factor M compared to the single-agent MTTF, i.e.,

$$\frac{\mathrm{MTTF}}{\mathrm{MTTF_{sys,max}}} = M \quad , \tag{10}$$

when comparing the results to [11]. This is intuitive because instead of only one agent potentially failing, there are M agents potentially failing.

A. Verification through System Level Simulation

Fig. 2 shows the results of extensive system-level simulation for the schemes S_1^0, S_2^0, and S_1^1 at $M = 20$ for $p_{\mathrm{loss}} = 30\,\%$. Each data point was simulated until at least one agent failed and the number of simulation runs was 10^6, therefore resulting in 10^6 $\mathrm{MTTF_{sys}}$ values for each data point. The simulated $\mathrm{MTTF_{sys}}$ values are shown as colored markers and for comparison, the closed-form $\mathrm{MTTF_{sys}}$ from (5) is shown as stair plot. The 99 % confidence intervals determined via the *large sample confidence interval* method is also plotted (in black). They are barely visible because the simulations match the analytical results very well.

B. SARA Schemes

Henceforth, we consider a per-channel packet loss probability of 10 %. Fig. 3 illustrates the analytical $\mathrm{MTTF_{sys}}$ versus the amount of available channels L_{av} among all resource allocation schemes of this article for $M = 20$. Please note the logarithmic ordinate axis. Human-readable orders of magnitude in time are also displayed.

As expected, each resource allocation scheme features an $\mathrm{MTTF_{sys}} = 180\,\mathrm{ms} = (K + 1) \times T_{\mathrm{s}}$ for $L_{\mathrm{av}} \to 0$ because at least one agent is not allocated any channel during this time and, thus, passes straight through its individual Markov chain (see Fig. 1), reaching the *down* state in the shortest possible time. On the other extreme, for $L_{\mathrm{av}} \to \infty$, the $\mathrm{MTTF_{sys}}$ reaches its maximum value (displayed with a colored dashed line), which can be calculated through (10).

Resource allocation schemes that feature a high number of base channels, i.e., the number of requested channels after a successful transmission, feature a weaker increase in the region $10 \le L_{\mathrm{av}} \le 20$ compared to schemes with a low number of base channels. This is because in this region the number of requested channels exceeds the number of available channels for all schemes and due to the random admission control as introduced in Sec. IV-B, the likelihood of agents ending up with 0 assigned channels is higher in schemes with a high base number. However, the strong increase in terms of $\mathrm{MTTF_{sys}}$ for S_1^0 is mitigated by the low $\mathrm{MTTF_{sys,max}} \approx 1\,\mathrm{min}$, which is clearly not useful in most closed-loop control applications.

The SARA schemes S_1^1 and S_2^1 outperform static MC by far. Comparing S_1^1 and S_2^0 (static dual-connectivity) for example, multiple advantages stand out:

1) The $\mathrm{MTTF_{sys,max}}$ with S_1^1 features a 100x improvement, which also complies with [11] and (10).
2) The concern of SARA's potentially poor applicability to a multi-user system with limited resources, which is the main motivator for this article, proves to be ill-founded. Reaching the identical $\mathrm{MTTF_{sys}}$ requires fewer available channels L_{av} for S_1^1. For $M = 20$, S_2^0 reaches its $\mathrm{MTTF_{sys,max}} \approx 3$ days at $L_{\mathrm{av}} = 40$, while the same $\mathrm{MTTF_{sys}}$ is reached for S_1^1 at only $L_{\mathrm{av}} = 23$.
3) Comparing the respective $\mathrm{MTTF_{sys,max}}$, S_1^1 also proves to scale better. At $M = 20$, the 99th percentile ($0.99 \times \mathrm{MTTF_{sys,max}} \approx 9$ months) is reached at $L_{\mathrm{av}} = 29$ for S_1^1 and therefore earlier than for S_2^0 ($\mathrm{MTTF_{sys,max}} \approx 3$ days) at $L_{\mathrm{av}} = 40$. That is, the 100x $\mathrm{MTTF_{sys}}$ increase requires only approximately $29/40 \approx 73\,\%$ of the channels at $M = 20$.

These advantages apparent from the diagram are complemented by the fact that at their respective $\mathrm{MTTF_{sys,max}}$, the number of channels that are assigned on average is only 55 % when comparing S_1^1 and S_2^0, i.e., approximately half [11]. As a general advantage, all static MC schemes of this article exhibit strictly convex curves until $\mathrm{MTTF_{sys,max}}$ is reached, which translates to an increased relative benefit for each additional channel in the system. On the other hand, for S_1^1, the curve saturates more smoothly towards reaching $\mathrm{MTTF_{sys,max}}$, which means that we can further reduce L_{av} without loosing orders of magnitude in terms of $\mathrm{MTTF_{sys}}$ quickly. For instance for S_1^1, when reducing L_{av} from $29 \to 27$, the $\mathrm{MTTF_{sys}}$ drops from $0.99 \times \mathrm{MTTF_{sys,max}}$ to $0.93 \times \mathrm{MTTF_{sys,max}}$ whereas for S_2^0 (reducing L_{av} from $40 \to 38$) it drops from $\mathrm{MTTF_{sys,max}}$ to $0.06 \times \mathrm{MTTF_{sys,max}}$, i.e., more than one order of magnitude.

C. Channel Utilization

The diagram in Fig. 4 shows for each L_{av} the time fraction of residing in any state S_i that utilizes all L_{av} channels. It is obvious that the 31st (for S_1^0), 61st (for S_2^0), and 91st (for S_3^0) channel do not offer any benefit to the system because they are never requested, resulting in a step drop of η. As expected, the

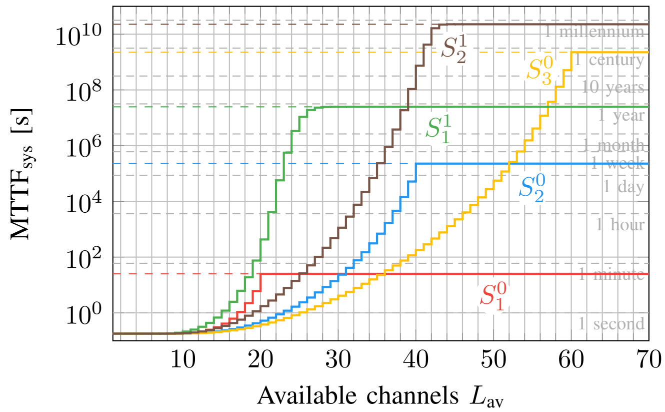

Figure 3. MTTF$_{sys}$ development for $M = 20$ agents.

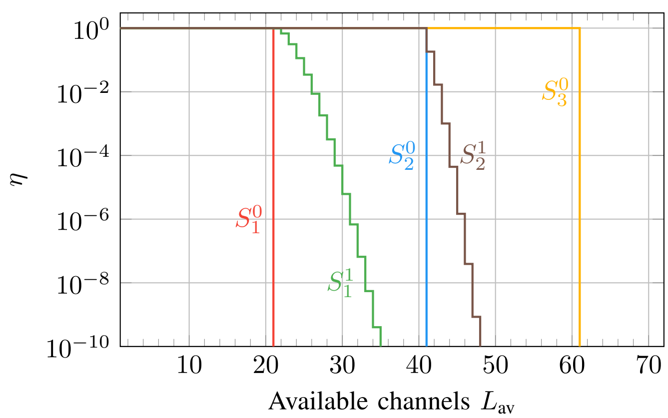

Figure 4. The channel utilization for $M = 20$.

dynamic schemes result in graphs that feature a more gradual decrease, indicating that additional channels are used by some (but not by all) system states. In other words, for the dynamic resource allocation schemes, some system states do not fully utilize all available resources, therefore wasting them. E.g., if all agents are in s_0, they request $L_{\text{req}}(S_0) = 20$ channels. If $L_{\text{av}} = 25$, 5 channels are wasted. For S_1^1 it was determined that the 99th percentile of MTTF$_{\text{sys}}$ is reached at $L_{\text{av}} = 29$, however, only approx. $0.1\,\%$ of all states utilize all 29 available channels. This leaves room for further improvement through randomly assigning unrequested channels to agents. This will further increase the maximum MTTF$_{\text{sys}}$ and also optimize the utilization of available channels, however, this is out of the scope of this article and will be left for future work.

VI. Conclusion

This article developed and verified a closed-form solution for deploying SARA on a system level, verified through system-level simulation. It demonstrates that using closed-loop control applications in conjunction with networks that use SARA is highly feasible in terms of the amount of channels required, also in a multi-user system with competition for resources. Control applications that are able to tolerate 3 consecutive packet losses strongly benefit from the S_1^1 scheme, which at 20 agents in the system only requires 1.4 channels per agent to reach $99\,\%$ of the maximum MTTF$_{\text{sys}}$, yielding a system-wide mean time to failure of approximately 1 year until the first agent fails. This constitutes an improvement of 100x compared to static dual-connectivity while only consuming 1.4 channels per agent instead of 2. In future works, further optimization can be conducted. For instance, when the number of available channels increases, increasingly many system states do not use all available channels when SARA is employed. These channels may be given these spare channels "on top" of requested channels, as otherwise they would be wasted. This will also impact the MTTF$_{\text{sys}}$, potentially increasing it beyond MTTF$_{\text{sys,max}}$ of this article. Furthermore, randomly denying agents' requested channels might not be optimal and other admission control schemes should be investigated, e.g., preferring agents that are close to failing.

Acknowledgments

This work was in part funded by the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany's Excellence Strategy – EXC 2050/1 – Project ID 390696704 – Cluster of Excellence "Centre for Tactile Internet with Human-in-the-Loop" (CeTI) of Technische Universität Dresden. This work was also in part sponsored by the Federal Ministry of Education and Research within the program "Twenty20 - Partnership for Innovation" under contract 03ZZ0528E - "fast robotics" and co-financed by public funding of the state of Saxony/Germany. We thank the Center for Information Services and High Performance Computing (ZIH) at TU Dresden for generous allocations of computer time. The authors alone are responsible for the content of the paper.

References

[1] German Federal Ministry of Education and Research, *Industrie 4.0*. Bundesministerium für Bildung und Forschung (BMBF), 2017.

[2] N. Ahmed, A. Mansoor, and R. Ahmad, "Mission-Critical Machine-Type Communications: An Overview and Perspectives towards 5G," *IEEE Access*, Jan. 2019.

[3] 3rd Generation Partnership Project, "Specification # 22.261 - Service requirements for next generation new services and markets," 3rd Generation Partnership Project, Technical Specification, 2019, version 16.9.0.

[4] Beckhoff Automation GmbH, "EtherCAT – The Ethernet Fieldbus," Beckhoff Automation GmbH, Tech. Rep., Nov. 2018.

[5] R. Abreu, P. Mogensen, and K. I. Pedersen, "Pre-Scheduled Resources for Retransmissions in Ultra-Reliable and Low Latency Communications," in *Proc. IEEE WCNC*, Mar. 2017.

[6] A. Avranas, M. Kountouris, and P. Ciblat, "Energy-Latency Tradeoff in Ultra-Reliable Low-Latency Communication With Retransmissions," *IEEE J. Sel. Areas Commun.*, Nov. 2018.

[7] W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems," *IEEE Control Syst. Mag.*, Feb. 2001.

[8] J. Hespanha, P. Naghshtabrizi, and Y. Xu, "A Survey of Recent Results in Networked Control Systems," *Proc. IEEE*, 2007.

[9] L. Zhang, H. Gao, and O. Kaynak, "Network-Induced Constraints in Networked Control Systems—A Survey," *IEEE Trans. Ind. Informat.*, Feb. 2013.

[10] G. N. Nair, F. Fagnani, S. Zampieri, and R. J. Evans, "Feedback Control Under Data Rate Constraints: An Overview," *Proc. IEEE*, Jan. 2007.

[11] L. Scheuvens, P. Schulz, T. Hößler, N. Franchi, A. Noll Barreto, and G. P. Fettweis, "State-Aware Resource Allocation for Wireless Closed-Loop Control Based on Multi-Connectivity," in *Proc. IEEE GLOBECOM*, Dec. 2020.

[12] T. Hößler, P. Schulz, M. Simsek, and G. P. Fettweis, "Mission Availability for Wireless URLLC," in *Proc. IEEE GLOBECOM*, Dec. 2019.

[13] J. Östman, R. Devassy, G. Durisi, and E. Uysal, "Peak-Age Violation Guarantees for the Transmission of Short Packets over Fading Channels," in *Proc. IEEE INFOCOM WKSHPS*, Apr. 2019.

[14] L. Scheuvens, T. Hößler, A. Noll Barreto, and G. P. Fettweis, "Wireless Control Communications Co-Design via Application-Adaptive Resource Management," in *IEEE 5G World Forum*, Oct. 2019.

[15] T. Hößler, L. Scheuvens, P. Schulz, A. Noll Barreto, M. Simsek, and G. P. Fettweis, "Dynamic Connectivity for Robust Applications in Rayleigh-Fading Channels," *IEEE Commun. Lett.*, Feb. 2020.

[16] A. Høyland and M. Rausand, *System Reliability Theory: Models and Statistical Methods*, ser. Wiley Series in Probability and Statistics. Wiley, 2009.

[17] C. Grinstead and J. Snell, *Introduction to Probability*. American Mathematical Society, 2012.

[18] G. F. Franklin, M. L. Workman, and D. Powell, *Digital Control of Dynamic Systems*, 3rd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1997.

[19] P. Wu and N. Jindal, "Coding versus ARQ in Fading Channels: How Reliable Should the PHY Be?" *IEEE Trans. Commun.*, Dec. 2011.