Hierarchical Management Algorithms for Highly Reliable Communication in 6G Industrial Environments

Published Version:
Hierarchical Management Algorithms for Highly Reliable Communication in 6G Industrial Environments / Borsatti, Davide; Davoli, Gianluca; Tonini, Federico; Cerroni, Walter; Raffaelli, Carla; Poe, Wint Yi; Trivisonno, Riccardo. - In: IEEE NETWORKING LETTERS. - ISSN 2576-3156. - ELETTRONICO. - 6:1(2024), pp. 10269136.36-10269136.40. [10.1109/LNET.2023.3321630]

Availability:
This version is available at: https://hdl.handle.net/11585/965566 since: 2024-03-07

Published:
DOI: http://doi.org/10.1109/LNET.2023.3321630

Hierarchical Management Algorithms for Highly Reliable Communication in 6G Industrial Environments

Davide Borsatti, Gianluca Davoli, Federico Tonini, Walter Cerroni, Carla Raffaelli, Wint Yi Poe, Riccardo Trivisonno

***Abstract*—This work explores the challenges of reliable communication in industrial applications, where human and robot collaboration requires highly reliable communication channels. We propose a hierarchical and distributed approach for solving the dynamic selection of resources. We assign distributed entities to portions of the managed working environment and investigate how to deal with users at their borders, proposing a mechanism to balance performance and resource efficiency. We support our analysis with numerical results obtained by simulation. By properly tuning the algorithm parameters, we can reduce energy consumption while satisfying the same reliability requirements.**

***Index Terms*—6G, adaptive reliability, low latency**

I. Introduction

Future mobile networks are expected to support services with extreme capacity, reliability, and latency requirements, well beyond the capabilities offered by the current 5G deployments [1]. This is particularly true in the case of advanced industrial environments, where moving robots can share the same working space as human operators, with the consequent critical risks in terms of safety in the working environment. To avoid collisions in these situations, the relative positions of humans and robots must be constantly monitored and, in case of potential impact, specific actions must be put in place within strict latency constraints. In a previous study, we discussed the specific challenges that this kind of use cases pose to the mobile network infrastructure, showing that the requirements in terms of communication reliability cannot be satisfied with the current features offered by 5G standards [2]. Not even the Dual Connectivity proposed for Ultra-Reliable Low-Latency Communication (URLLC) services seems to be sufficient [3].

From another perspective, increasing the number of redundant communication channels to achieve the target reliability levels requires additional resource deployment and, as a consequence, a higher overall network energy consumption. In addition, keeping multiple redundant connections active all the time could reduce the number of different devices that can be served by a given cell. Approaches based on dynamic resource allocation have been proposed in other fields, e.g. security [4]. The idea is to dynamically provision only the resources that are actually needed, improving the resource and energy consumption. In line with this approach, we developed the idea of *dynamic reliability*, i.e., an adaptive approach such that the network control plane is able to differentiate between critical and non-critical operating conditions and enforce the stringent reliability requirements only when needed [2].

To be effective in terms of guaranteed safety, such an adaptive management of redundant communications requires fast reaction times. The identification of a critical condition when a human operator is getting close to a moving robot, as well as the decision on the amount of redundant resources that must be deployed to satisfy the prescribed reliability requirements, must happen in a very short time frame. This could pose additional scalability challenges when the number of Access Nodes (ANs) that provide connectivity and the number of devices to be connected increase. In fact, this has an impact on the complexity of the problem to solve, with a consequent expected increase in the time needed to manage the adaptive channel assignment. Therefore, we explored the possibility to distribute the algorithms for critical situations detection and redundant channel selection, resulting in reduced execution times [5]. The main idea is to apply the adaptive approach to a hierarchical and distributed framework, splitting the area of the factory floor into sub-areas. Each sub-area is assigned to a distinct distributed computation element, which sees a reduced problem space and can then make decisions within a decreased execution time.

However, any approach that aims at distributing control plane functions introduces the issue of maintaining consistent information to make optimal decisions. In the use case we are considering here, this is reflected by the problem of managing users located at the boundaries of the factory floor sub-areas, where the distributed control element in charge of a given sub-area is not aware of whether there are moving robots in an adjacent sub-area that are critically close to the boundary. Existing work deals with the problem of improving performance for users located at the border by handling clusters of cells with distributed elements (e.g., [6] and references therein). Even though the existing solutions exhibit good performance gains, they focus on minimizing the interference for users rather than imposing a (stringent) target on reliability. In

This work has been funded by the Huawei-CNIT Joint Innovation Center (JIC).

Davide Borsatti, Gianluca Davoli, Walter Cerroni, and Carla Raffaelli are with the University of Bologna, Department of Electrical, Electronic and Information Engineering "Guglielmo Marconi," and with the National Inter-University Consortium for Telecommunications (CNIT), National Laboratory of Wireless Communications (WiLab), Bologna, Italy.

Federico Tonini is with the National Inter-University Consortium for Telecommunications (CNIT), National Laboratory of Wireless Communications (WiLab), Bologna, Italy.

Wint Yi Poe and Riccardo Trivisonno are with Huawei Technologies, Munich, Germany.

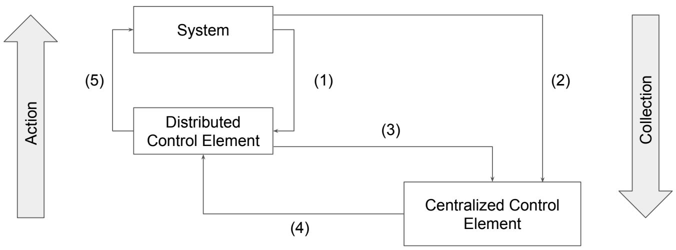

Fig. 1. Hierarchical framework composed of Distributed and Centralized Control Elements, with reference points.

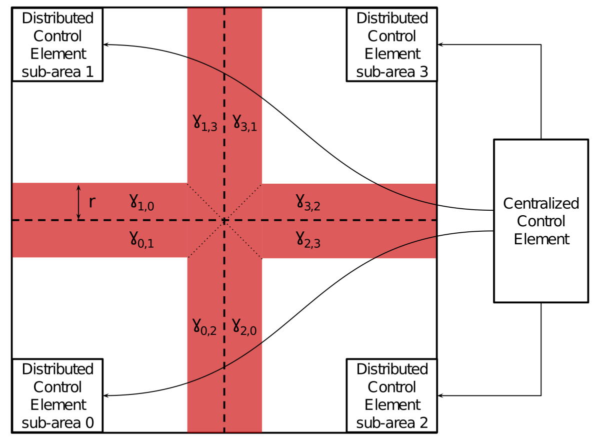

Fig. 2. Representation of a factory floor split into four sub-areas (dashed black lines), each associated with an index $\{0, 1, 2, 3\}$. Each sub-area has its own "Distributed Control Element", all controlled by the "Centralized Control Element". The area within distance r from the borders is highlighted in red.

our previous study, we adopted a conservative approach, i.e., to provide always the highest possible reliability to users located at the sub-area boundaries [5]. However, this leads to excessive resource consumption and sub-optimal redundant channel allocation.

In this paper, we extend the work in [5] by proposing a strategy to reduce resource consumption at the sub-area borders with a distributed control architecture supporting dynamic reliability. The strategy relies on a probability of activating link redundancy for users at the borders for each distributed element, while a centralized component tunes these probabilities based on the current crowding of the different sub-areas. We support our proposal with numerical data gained through simulation. We demonstrate in detail how, by fine-tuning the algorithm parameters, we may decrease resource overprovisioning while preserving the service reliability requirements.

II. Hierarchical Management of Dynamic Reliability

We consider an industrial environment, represented by a large factory floor, where moving robots operate alongside humans. Each robot and each human carries User Equipment (UE) and is free to move around the working area, covered by a set of ANs. When a robot gets too close to a human or to another robot, the network should provide its UE with a highly reliable connection in order to control the machine more accurately or stop it, if necessary. This proximity situation is identified when the detected distance between two UE devices is below a given value r called *critical distance*.

Providing high communication reliability requires to dynamically assign multiple bearers to UEs. This can be done in a distributed fashion, reducing the execution time [5]. However, the current 5G architecture is not designed to support such a distributed scenario. More flexible architectures for 6G networks supporting verticals with extreme requirements are under investigation [7], [8], while studies on network requirements for stable operations of service robots are ongoing [9]. In this work, we adopt the two-level hierarchical control framework depicted in Fig. 1. In the figure, the system block represents a generic functional element of the network architecture under control. The control logic is split into two levels: a distributed element performs fast and system-specific actions, while a centralized one has a broader scope/view and is in charge of defining/refining the policies for the distributed element behavior. The interconnections from an internal block to an external one (denoted with 1, 2, 3) are introduced to implement data analytics and allow the control logic to gather the necessary information to produce appropriate actions to be enforced. All interconnections in the opposite direction (4 and 5) represent the enforcement of such actions. Such a hierarchical framework is applicable to a distributed scenario, where the overall system is divided into multiple parallel subsystems, each controlled by one distributed element, all connected to the same centralized element.

Figure 2 shows an example of a factory floor divided into four sub-areas, where each distributed control element performs control actions on the UEs moving in the related sub-area. More specifically, it identifies a suitable combination of radio bearers to be established with each UE in its sub-area that is in critical distance conditions. The set of redundant radio bearers B_c^u to be established with device ue_u, for each ue_u in critical conditions, must ensure that the combined loss probability:

$$\ell_u = \prod_{b=1}^{|B_c^u|} \ell_b^u \,, \tag{1}$$

is such that $\ell_u \leq \ell_d$, where $|B_c^u|$ is the number of bearers in the candidate set for ue_u, ℓ_b^u is the loss probability offered to ue_u by bearer b in the candidate set B_c^u, and ℓ_d is the target loss probability determined by the required level of reliability.

The objective of the algorithm is to minimize the number of radio resources employed while satisfying the constraint on the loss rate. While the solution to this problem is a task for each distributed component, the centralized element may decide how many sub-areas the whole scenario should be divided into, the reliability level required in a given sub-area, and the set of *border policies* to be enforced in the sub-areas. Splitting the factory floor into sub-areas, each assigned to an independent distributed component, introduces the problem of managing UEs that are close to the sub-area borders, i.e., within the red bands shown in Fig. 2. Each distributed component is only aware of UEs inside its assigned area, so a UE that finds itself in the proximity of a sub-area border might be within critical distance from another UE in a neighboring area, with the respective distributed entities being unaware of it.

For this reason, the centralized component may enforce a policy to manage these situations, by associating each border between sub-areas i and j to the probability $\gamma_{i,j}$ of activating link redundancy for UEs that are within critical distance to that border. Setting $\gamma_{i,j} = 1$ means that the distributed components always try to enforce high reliability for UEs located on the border between sub-areas i and j, regardless of the presence of other UEs in the neighboring area. Conversely, with $\gamma_{i,j} = 0$ the distributed component never associates redundant paths to UEs close to that border, unless the UE is also close to another one in the same sub-area. The former is the conservative policy assumed in [5], resulting in excessive resource consumption due to redundant radio links established also when the UEs do not actually need them. On the other hand, enforcing the latter policy might result in reduced operation reliability, as UEs that would require redundancy are not considered in critical conditions. The centralized entity could then tune these probabilities at runtime for each sub-area border of each distributed component, based on information such as the number of UEs in a neighboring sub-area, their expected mobility, or other metrics.

Given the methodology considered, a UE located near the border of a sub-area can either be within critical distance of another UE in a neighboring sub-area or not, and it can either be provided with a reliable connection through redundancy or not. Given N^s_{bor} as the total number of UEs on sub-area s borders, we have $N^s_{bor} = N^r_c + N^{nr}_c + N^r_{nc} + N^{nr}_{nc}$, with N representing a number of UEs, the subscripts c and nc representing the fact that the UE is within or not within critical distance of another UE in a different sub-area, and the superscripts r and nr representing the fact that the UE has been or has not been assigned redundant connection paths. In particular, two of these terms are relevant to assess the performance of the distributed algorithm: N^{nr}_c, which counts the number of UEs that would need a higher reliability level but that did not obtain it from the resource allocation mechanism, and N^r_{nc}, which counts the number of UEs that have been assigned more resources than needed. By normalizing these terms over N^s_{bor}, we obtain percentages that represent an *underprovisioning* and an *overprovisioning rate* of the algorithm, respectively.

Ideally, the centralized component should choose the right set of border probabilities $\mathbf{\Gamma} = \{\gamma_{i,j}\}$ such that the following quantity is minimized:

$$\min_{\mathbf{\Gamma}} \ \alpha N^{nr}_c + (1-\alpha) N^r_{nc} \tag{2}$$

As the underprovisioning and overprovisioning rates are inherently in opposition, the centralized component should aim at balancing the contribution of the two terms to the total sum of UEs. This design freedom in the management of the system is represented by the parameter α, that might be tuned at runtime. Intuitively, with $\alpha = 1$ the centralized component targets a complete reduction of the *underprovisioning* rate, neglecting the resources wasted by *overprovisioning* some UEs. Conversely, with lower values of α, the centralized control element tries to lower the *overprovisioning* rate while keeping bounded the *underprovisioning* one.

Algorithm 1: Dynamic bearer allocation in sub-area s

Input : critical distance r; acceptable loss probability ℓ_d; set of UEs and ANs in the sub-area $U = \{ue_u \mid u = 1, \ldots, N^s_U\}$ and $A = \{an_a \mid a = 1, \ldots, N^s_A\}$, limits on the number of radio bearers that can be associated to a single UE $\overline{N}^u_b$ and AN $\overline{N}^a_b$, $\forall\, ue_u \in U$, $\forall\, an_a \in A$.

Identify UEs in critical distance conditions
$C,\, C_b \leftarrow ComputeCriticalSets(U)$
Compute the required bearer set for each UE
$[B^u_c] \leftarrow ComputeBearerAllocation(C, C_b)$

Algorithm 2: $ComputeCriticalSets(U)$

Build set P of UE pairs
$P = \{(ue_x,\, ue_y) \mid \forall\, ue_x, ue_y \in U,\ ue_x \neq ue_y\}$
Initialize set C of UE in critical distance conditions
$C \leftarrow \emptyset$
foreach UE pair $(ue_x, ue_y) \in P$ **do**
  if UE pair is Human-Robot or Robot-Robot **then**
    # Compute distance between UEs
    $d(t) \leftarrow dist(ue_x, ue_y)$ # Euclidean distance
    if $d(t) \leq r$ **then**
      if $ue_x \notin C$ **then**
        Add ue_x to C and remove it from U
      if $ue_y \notin C$ **then**
        Add ue_y to C and remove it from U

Initialize set C_b of UEs near the border that will have a higher reliability level
$C_b \leftarrow \emptyset$
foreach $ue_u \in U$ **do**
  # Iterate over other sub-areas
  forall $n \in \{0, \ldots, N_{sa} - 1\} \mid n \neq s$ **do**
    if $\gamma_{s,n} > 0$ **then**
      # Compute distance from the border with sub-area n
      $d^n(t) \leftarrow dist(ue_u, border_n)$
      if $d^n(t) \leq r$ **then**
        # Sample from uniform distribution in [0, 1)
        if $random(0,1) \leq \gamma_{s,x}$ **then**
          # Add ue_u to C_b
          $C_b \leftarrow C_b \cup \{ue_u\}$
          break

III. Distributed Redundant Path Assignment

The distributed component in charge of sub-area s, with $s = 0, \ldots, N_{sa} - 1$ and N_{sa} being the number of sub-areas, tackles the problem described in the previous section by running Algorithm 1 on the set of UEs located in sub-area s. The algorithm consists of two main parts: the former analyzes the current situation and determines which UEs require highly reliable connectivity, while the latter tries to achieve it by looking for a suitable radio bearer set for each of those UEs.

Algorithm 2 details how we build the set of UEs in critical distance conditions (C). UEs are paired, considering all possible combinations once, and those whose distance is lower than the threshold r are individually added to the critical set C and removed from U. Afterward, the algorithm checks whether some of the remaining UEs in U are at a distance smaller than or equal to r from the sub-area border. If the condition

Algorithm 3: $ComputeBearerAllocation(C, C_b)$

Randomize item order in C and in C_b
foreach $ue_u \in C \cup C_b$ **do**
  # Initialize bearer allocation set for ue_u
  $B_c \leftarrow \emptyset$
  Estimate loss rate ℓ_j with just one bearer b_j toward each Access Node an_a whose number of used bearers is $N^a_b < \overline{N}^a_b,\ \forall a \in A$
  Construct ordered set B_O of bearers sorted by loss rate from best bearer b_1 associated to the lowest loss rate ℓ_1 to worst bearer $b_{N^s_A}$ associated to the highest loss rate $\ell_{N^s_A}$
  $k \leftarrow 1$
  while $B_c = \emptyset$ and $k \leq \overline{N}^u_b$ **do**
    Build $B^k_c \leftarrow \{b_1, \ldots, b_k\}$ of cardinality k including the first k elements of B_O
    Compute estimated combined loss rate ℓ_k yielded by candidate set B^k_c
    $\ell_k \leftarrow \prod_{j=1}^{k} \ell_j$
    if $\ell_k \leq \ell_d$ **then**
      $B_c \leftarrow B^k_c$
      break
    else
      $k \leftarrow k + 1$
    if $B_c \leftarrow \emptyset$ **then**
      Report failure to centralized component
    else
      Store bearer allocation set B_c for UE ue_u at simulated time t and update the number of used bearers N^a_b for relevant Access Nodes

is verified, then the UE is added to the *critical border* set C_b with a given probability γ, depending on the current *border policy* configuration. UEs in sets C and C_b are provided with highly reliable connectivity.

Algorithm 3 shows how we try to identify the smallest set of radio bearers such that the combined estimated loss rate ℓ_u perceived by each UE in critical conditions stands below a certain desired level ℓ_d. To achieve this goal, the algorithm considers one UE in critical conditions at a time, builds a list of available bearers, sorts them by provided loss rate (best to worst), then picks the first best one(s) until the desired level of overall loss rate is met, or no suitable combination is available. In this way, considering the definition of combined loss rate given in Eq. (1), if the algorithm succeeds, we are guaranteed to have found the smallest possible set of bearers that satisfies the constraints.

IV. Case Study and Numerical Evaluation

To validate the algorithm and to show how tuning the $\mathbf{\Gamma}$ probabilities can impact the performance of the system, we developed a software simulator, written in Python, based on the one employed in [5]. We ran it on a Ubuntu 20.04 virtual machine with 8 vCPUs and 8 GB of RAM. The simulator allows to reproduce a variety of operating conditions. Specifically, we consider a square factory floor like the one represented in Fig. 2, with sides of $100\,m$, $N_A = \sum_{s=0}^{N_{sa}-1} N^s_A = 100$ ANs deployed across it, and a total of $N_U = \sum_{s=0}^{N_{sa}-1} N^s_U \in \{70, 100\}$ UEs moving on it. For the estimation of the Block

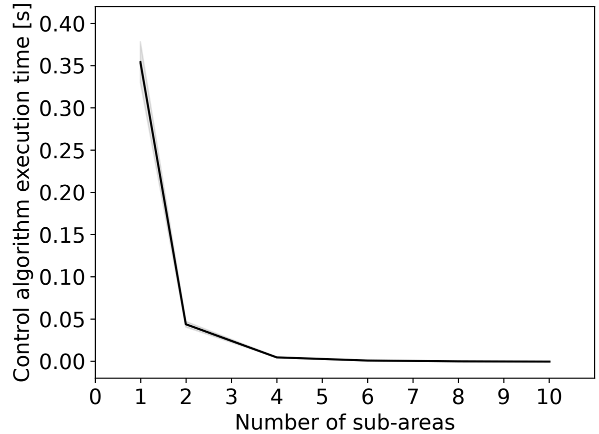

Fig. 3. Control algorithm execution time as a function of the amount of sub-areas, with $\ell_d = 10^{-5}$. A 95% confidence interval is shown in grey around the black line.

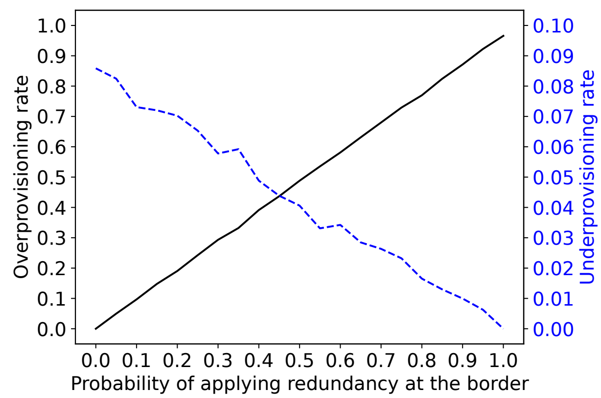

Fig. 4. Overprovisioning (solid line) and underprovisioning (dashed line) rate as a functions of the probability of applying redundancy to UEs near the border, when UEs are distributed evenly and the same probability γ is applied on every border.

Error Rate (BLER), we follow the guidelines given by 3GPP in TR 38.901 V17.0.0 [10] for radio channels between 0.5 GHz to 100 GHz as explained in [2]. The estimation would benefit from the use of more refined radio models, but this is not in the scope of this paper. We considered a desired loss rate level of $\ell_d = 10^{-5}$, as an intermediate value between different relevant use cases considered in [11].

First of all, we show that by splitting the scenario into sub-areas we can achieve a substantial improvement in terms of the execution time of the control algorithm, which decreases sharply with the increase of the number of sub-areas, as shown in Fig. 3, proving the benefits of the architectural choice. This improvement is due to the reduced combinations UEs-ANs among which the distributed elements need to search.

After fixing the number of sub-areas to $N_{sa} = 4$, we distribute the UEs and ANs evenly across them, i.e., $N^s_U = N^s_A = 25$, $s = \{0, 1, 2, 3\}$, and we investigate the performance in terms of underprovisioning and overprovisioning, as previously defined, over the scenario depicted in Fig. 2, for a range of values of probability γ, applied such that $\gamma_{i,j} = \gamma\ \forall i, j$. Figure 4 shows that the underprovisioning

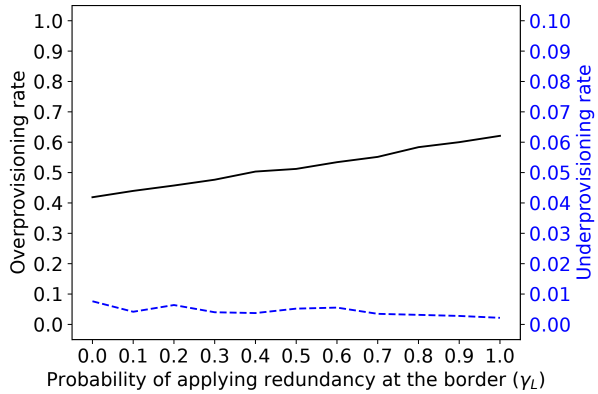

Fig. 5. Performance when UEs are distributed unevenly and the probability of applying redundancy at the border between most crowded sub-areas if fixed to $\gamma_H = 1$, varying γ_L.

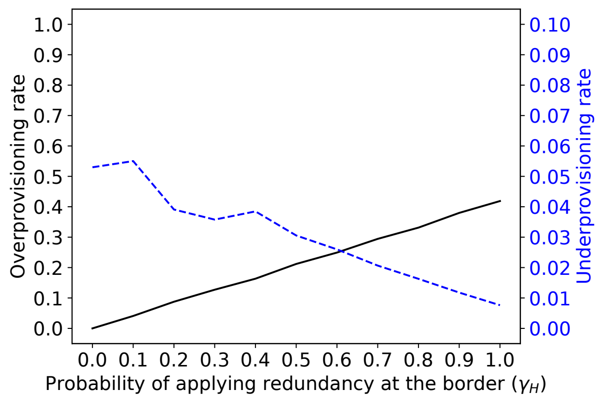

Fig. 6. Performance when UEs are distributed unevenly and the probability of applying redundancy at the border between less crowded sub-areas if fixed to $\gamma_L = 0$, varying γ_H.

and overprovisioning of resources to UEs at the border are in opposition, as expected. In fact, the best performance in terms of underprovisioning (i.e., all UEs are provided with a reliable connection via redundancy) correspond to the worst performance in terms of overprovisioning (i.e., most UEs that are provided with a reliable connection do not actually need it), and vice versa.

Next, we redistribute the UEs over the sub-areas depicted in Fig. 2 to create an imbalance, i.e., $N_U^0 = 5$, $N_U^1 = 25$, $N_U^2 = 0$, $N_U^3 = 40$, and we examine the performance resulting from applying different probabilities γ on different borders, based on the number of UEs present in each of the sub-areas. We define the probability of applying redundancy to UEs at the border between the two most crowded sub-areas ($s = 1$ and $s = 3$) as $\gamma_{1,3} = \gamma_{3,1} = \gamma_H$, and the probability at the border between less crowded areas ($s = 0$ and $s = 1$) as $\gamma_{0,1} = \gamma_{1,0} = \gamma_L$. We set all other probabilities to $\gamma = 0$, including those towards the area with no UEs in it.

In Fig. 5 we can observe how γ_L has almost no impact on the underprovisioning rate, therefore keeping γ_L low helps reduce the overprovisioning of resources without significant performance degradation. This behavior is in line with the expectations, as it is not necessary to provide highly reliable connections to UEs that are unlikely to be close to those on the other side of the border if the neighboring sub-area is scarcely crowded. In Fig. 6, we can observe that γ_H has a substantial impact on the overprovisioning rate, but it also affects the underprovisioning rate, as with low values of γ_H many UEs are not provided with redundant access even though they would require it. Likewise, this behavior was expected since the two adjacent sub-areas have a high user density. Overall, the centralized component should tune γ_H and γ_L to obtain the lowest possible underprovisioning rate while also keeping the amount of overprovisioned users limited.

V. Conclusion

In this paper, we showed how a general hierarchical approach that we foresee for future mobile networks could enable the support of dynamic reliability levels, depending on the changing application needs. Simulation results established the gains of this approach in terms of execution time. Furthermore, they showed how the centralized element could change the performance of the system by varying a few parameters, enforcing stricter reliability in dense areas while decreasing resource waste in uncrowded areas. In future work, we would like to extend this approach by adding AI to the proposed framework to improve the performance of the algorithms.

References

[1] 5G-PPP - The 6G Architecture Landscape - European perspective. [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2022/12/6G-Arch-Whitepaper_v1.0-final.pdf

[2] D. Borsatti *et al.*, "Adaptive network reliability for human-robot interaction in beyond 5G industrial applications," in *2022 18th International Conference on the Design of Reliable Communication Networks (DRCN 2022)*, 2022.

[3] 5G; System architecture for the 5G System (5GS) (3GPP TS 23.501 version 16.6.0 Release 16)). [Online]. Available: https://www.etsi.org/deliver/etsi_ts/123500_123599/123501/16.06.00_60/ts_123501v160600p.pdf

[4] S. Shen *et al.*, "Adaptive and Dynamic Security in AI-Empowered 6G: From an Energy Efficiency Perspective," *IEEE Communications Standards Magazine*, vol. 5, no. 3, pp. 80–88, 2021.

[5] D. Borsatti *et al.*, "Distributed management of dynamic reliability in Beyond 5G industrial environments for extreme low latency communication," in *19th International Conference on the Design of Reliable Communication Networks (DRCN 2023)*, 2023.

[6] A. Zubow *et al.*, "On practical cooperative multi point transmission for 5G networks," *Computer Networks*, vol. 171, p. 107105, 2020.

[7] One6G group white paper - 6G technology overview. [Online]. Available: https://one6g.org/download/2001/

[8] D. Tsolkas and H. Koumaras, "On the Development and Provisioning of Vertical Applications in the Beyond 5G Era," *IEEE Networking Letters*, vol. 4, no. 1, pp. 43–47, 2022.

[9] 3GPP TR 22.916 version 0.1.0 - Study on Network of Service Robots with Ambient Intelligence (Release 19).

[10] 3GPP TR 38.901 version 17.0.0 - 5G; Study on channel model for frequencies from 0.5 to 100 GHz (Release 17).

[11] NIST Advanced Manufacturing Series 300-8, Revision 1 - Wireless User Requirements for the Factory Workcell. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ams/NIST.AMS.300-8r1-upd.pdf