Accepted for presentation in: 2023 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)

Distributed Realtime Wireless Network Emulation for Multi-Robot and Multi-Link Setup Evaluation

Manuel Patchou[1], Tim Gebauer[1], Cedrik Krieger[1], Stefan Böcker[1] and Christian Wietfeld[1]

***Abstract*— State-of-the-art robot deployments for rescue robotics typically involve more than one robot, each potentially equipped with multiple radio access technologies for redundancy. Evaluating and benchmarking such deployments with numerous robots and network communication links requires tools that scale well across these dimensions. This work builds on previous work that presented vSTING as a solution for emulating challenging network environments for single robots. The current work extends vSTING to include multi-robot and multi-link support, making it a pure software solution but distributed, eliminating the need for dedicated hardware. Furthermore, the resulting solution can be used in real-world applications to manage or limit network resources. In the first evaluation step, we validate the multi-robot and multi-link feature. Then, an exemplary validation of a basic multi-link solution in its early development stage is conducted by replaying a network environment recorded during a prior mission.**

I. INTRODUCTION

The range of capabilities expected of robots in rescue missions includes a variety of aspects, such as high mobility in difficult terrain, dexterity in manipulations such as opening doors, and aerial surveillance. Since it is impossible for a single robot to cover all these requirements, the use of multiple robots in rescue missions has become a common practice. In addition, the network environment and conditions in which rescue robots are deployed can change from mission to mission and even within the same mission. This motivates a multi-link approach, where multiple networks are used for communication redundancy by equipping the robots with multiple radio access technologies. Benchmarking and evaluating such multi-robot and multi-link setups requires solutions capable of handling the complexity introduced by the diversity of robots and network links. To this end, we propose an additional approach to virtual STING (vSTING), our solution for the evaluation of teleoperated mobile robots using real-time wireless network emulation, which introduces support for multiple robots and network links through a distributed approach. The contributions of this work can be summarized as shown in Fig. 1:

- **Multi-Robot** support
- **Multi-Link** support
- **Resource Management** of the robot fleet
- **Replay Orchestration** of recorded environments

The remainder of the paper is organized as follows: After discussing related work in Sec. II, the distributed vSTING concept is introduced in Sec. III followed by implementation aspects given in Sec. IV. Finally, evaluation results are discussed in Sec. V.

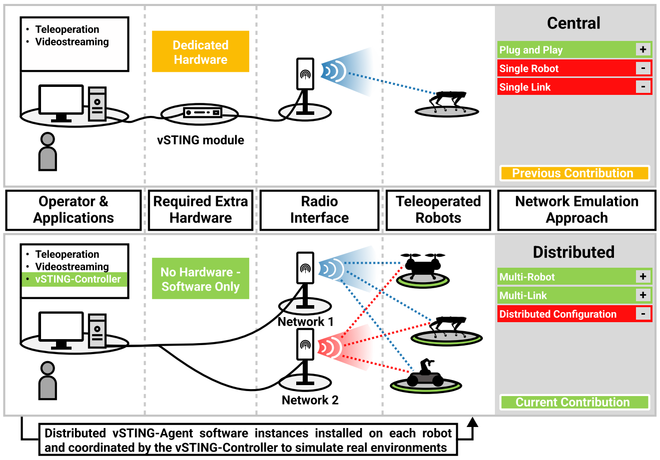

Fig. 1. The alternative approach envisioned for evaluating robots under realistic connectivity conditions: using a distributed system to remotely control network emulation for the entire robot fleet.

II. RELATED WORK

Robot assisted rescue missions now frequently and typically involve deploying multiple robots to address the wide range of challenges that may arise. In [1] a drone fleet was heterogeneously equipped for tasks ranging from transportation to modeling and situational awareness. Coordinating teams of robots often requires some sort of decentralized approach [2], even in setups with central decision making, since these decisions must be executed on distant systems. This stresses the critical character of network communications in robot teams even further. Radio communication tests are included in the suite of standard methods published by the National Institute of Standards and Technology (NIST) for evaluating emergency response robot capabilities in repeatable ways [3]. In the spirit of this test suite, the international Robocup Rescue competition follows a systematic approach [4] to evaluate the specific skills of robotic setups in the context of rescue missions. The vSTING module proposed in our previous work [5] that relies on network emulation to bring forth realistically challenging network environments for robot evaluations was integrated and used in the challenges of the 2023 edition of the RoboCup Rescue competition in Bordeaux. Impressions are shown in Fig. 2. The positive feedback of the vSTING integration at the

---

[1] Manuel Patchou, Tim Gebauer, Cedrik Krieger, Stefan Böcker, and Christian Wietfeld are with the Communication Networks Institute, TU Dortmund University, 44227 Dortmund, Germany {Manuel.Mbankeu, Tim.Gebauer, Cedrik.Krieger, Stefan.Boecker, Christian.Wietfeld}@tu-dortmund.de

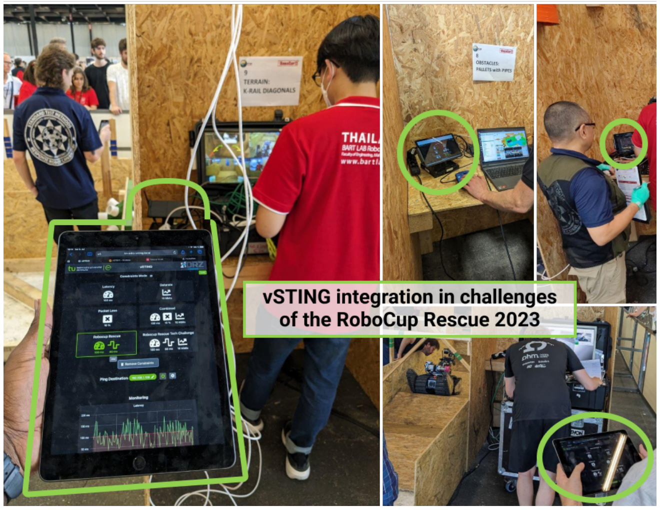

Fig. 2. vSTING during the International RoboCup Rescue 2023 Championship: vSTING was integrated into some arenas of the RoboCup Rescue League and teams that volunteered to challenge their robot's mobility under network constraints received extra points as a reward.

RoboCup stresses the importance of assessing network-reliant robotic systems and applications in degraded network environments. A compilation of such applications is provided in [6]. This assessment must also be application-aware and inspired from real-world scenarios and use-cases to complete the isolated unit tests of components [7].

The Assesment of a system's network behavior and capabilities in a targeted environment can be carried out in progressive steps. Often first in the solution development process, simulations allow for the preview and expectation check on possible and achievable communication performance in the targeted deployment environment. An example is given in [8] with a joint simulator for network and mobility being specialized for handling hybrid fleets of aerial and ground vehicles to cover use cases of drone-enabled parcel delivery. A futher step can then be made towards reality by bringing in some hardware into the software simulation process. Hardware in the Simulation Loop (HiSL) simulations can enhance the validation of simulation results. A general framework for building HiSL simulations was presented in [9], and demonstratively used to confirm the multi-Radio Access Technology (RAT) features of a multi-link communication module [10].

A higher level of validation of simulation results can then be achieved through experiments with real entities. In [11], a solution to measure the resilience of systems in networks with interfering traffic was proposed using spatially distributed real nodes to generate traffic.

Finally, due to costs and logistics tied to experimentation with real entities, this is not always a feasible option. Therefore simulating and emulating relevant aspects within a real entity is an optimal compromise to accelerate the feedback loop in iterative development processes. This train of thought was followed in [12] where network emulation was used to cost-effectively measure the impact of usual network constraints such as latency and packet loss on real-time teleoperated driving. This is the philosophy followed in [13] and detailed in the current work to provide a means to inspect and tune the behavior of robot teams using multiple network links by relying on distributed network evaluation.

III. THE DISTRIBUTED vSTING APPROACH

To address common scenarios involving multiple robots and network technologies, the vSTING concept of our previous work was evolved in a distributed manner to a remote network emulation. The architectural changes and resulting benefits are shown in Fig. 3. The original vSTING concept can be summarized as Traffic shaping consists of adapting the outgoing traffic to a specific rate by buffering, delaying or dropping packets as needed. While the strength of this approach lies in its modularity and ease of installation, as the existing network infrastructure remains unchanged and the vSTING module is simply connected like a switch between the operator and the robot, it encounters scalability limitations in multi-robot or multi-link configurations. In fact, each vSTING module can only serve one robot and one link at a time. This new concept, distributed vSTING, does not rely on additional hardware to artificially limit network traffic through network emulation. Network emulation is done directly on each robot for each of its network interfaces and is controlled locally by a software module: the vSTING-Agent. This results in a system requirement for the robots. They must use vSTING-compatible operating systems. For now, this means a system running a Linux distribution with the Linux kernel traffic control suite and special utilities. These utilities are required for traffic shaping and are covered in section III-A. All vSTING-Agents running on the individual robots are coordinated centrally by the vSTING-Controller, a software module running in the remote control center. The robots running the vSTING-Agents must therefore be reachable from the vSTING-Controller via at least one of the available network connections. Furthermore, existing features of the vSTING module being powered by distributed approach and the resulting multi-robot and multi-link support become more advanced. The link monitoring feature evolves in resource management, therefore allowing the usage of the distributed vSTING as a means of allocating network usage within the robot fleet, ensuring that no robot can monopolize the network at any given time, to the detriment of other robots' transmissions. The network replay feature, which allows to recreate network environments by using the traces recorded in these as network constraints can now be used for multiple robots and multiple links, thus evolving in a so called replay orchestration.

In order to support multiple network links and the possible connections that can be made by multiple robots over them, the way the Queueing Discipline (QDiscs) provided by the linux traffic control tool TC are used had to be changed as well. The main aspects of these improvements are shown in Fig. 4 and discussed in the following sections.

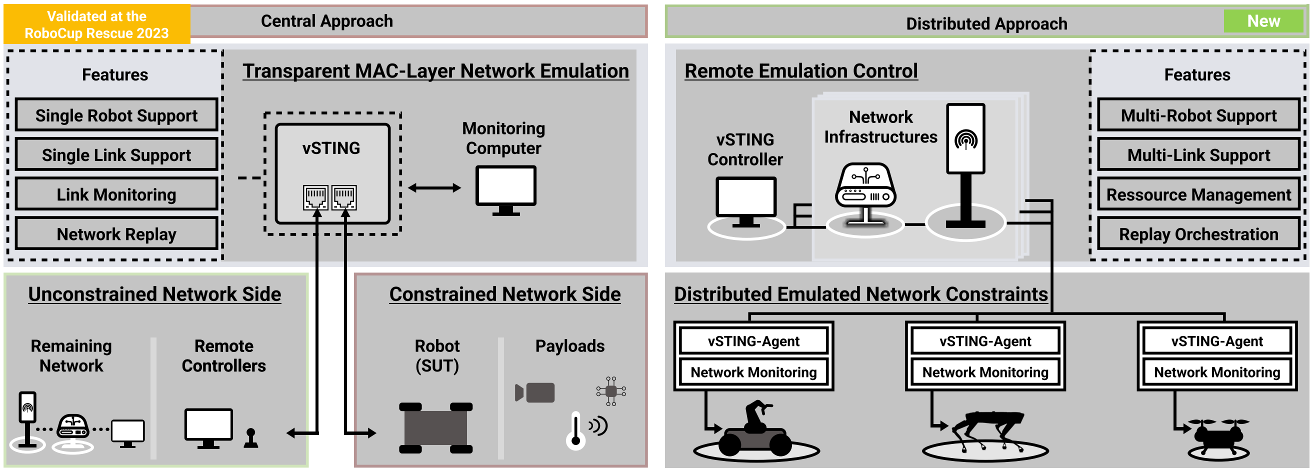

Fig. 3. Concept for the new vSTING approach to support multiple robots and networks: distributed network emulation. This allows the vSTING concept to scale across robots and networks. However, the central vSTING concept remains relevant through key features such as ease of installation.

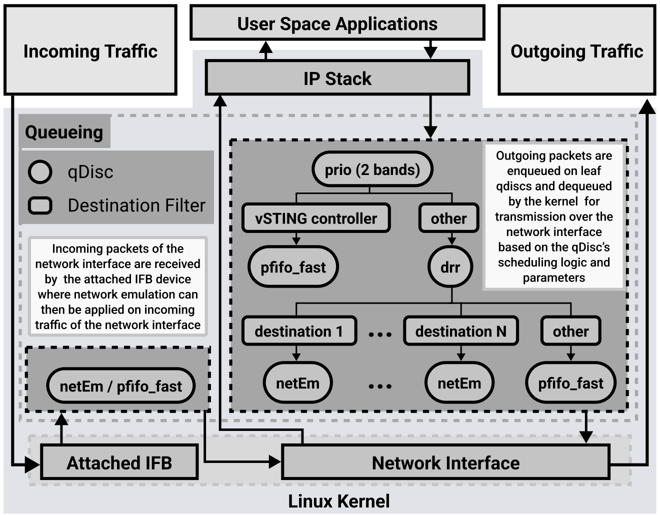

Fig. 4. The queueing structure used to achieve destination constraints alongside general ones. This architecture gives precedence to destination constraints over general constraints.

A. *Bidirectional Destination Based Traffic Shaping*

Considering a link represented by a connection between two robots. To bidirectionally enforce a given network constraint such as a delay of 100 ms on such a link, shaping of said delay would need to be applied at both ends of the link. Each robot should therefore apply shaping when sending packets to the other robot. Furthermore, this shaping should only be applied to packets destined for the specific robot at the other end of the link. Connections to other robots or to operators must remain unaffected. To achieve this behavior, the structure in the right queueing block shown in Fig. 4 is shown. To be transmitted, packets traverse the queueing structure until they reach a leaf node, where they are enqueued and wait for transmission by the kernel.

Following this structure, packets coming from user applications through the Internet Protocol (IP)-stack first arrive in the *prio* QDisc which then decides either to enqueue them for undelayed transmission in the *pfifo_fast* QDisc if they are meant for the network controller, or to forward them to the *drr* QDisc. This first branching ensures that the communication with the vSTING-Controller is always unaffected and will be further referred to as the *controller QDisc-branching*. Branching between destinations happens on the *drr* QDisc. If traffic towards a specific destination must be shaped, there must be a branch with an appropriate filter that matches the header of the IP packet to ensure that it has the correct destination. The packet is then enqueued at the leaf *netEm* QDisc of the so called *destination QDisc branch*. if no matching destination filter is found within the *destination branching* of the *drr* QDisc, the packet will follow the default branch named *other* where packets for which no specific shaping constraints exist are enqueued.

B. *General Interface Shaping*

In the event that a network constraint must be applied on all the traffic of a particular technology, i.e, that is going through or originating from a particular network interface, general interface shaping is required. This is a special case because incoming traffic must also be constrained, which by definition is outside the scope of traffic shaping which only considers outgoing traffic.

To circumvent this limitation, we use an Intermediate Functional Block (IFB), which is a pseudo network interface that acts as a concentrator for several different traffic sources. Packets coming from or destined to other interfaces can be redirected there to be processed using the *mirred* action. Incoming traffic meant for the network interface to be constrained can, therefore, be redirected to the IFB where shaping can then be applied using the *netEm* QDisc as it leaves the IFB for the network interface. The treatment of incoming traffic is shown in the left queueing block of Fig. 4. The traffic of the IFB is also subjected to a *controller branching*. This summarizes how general interface shaping

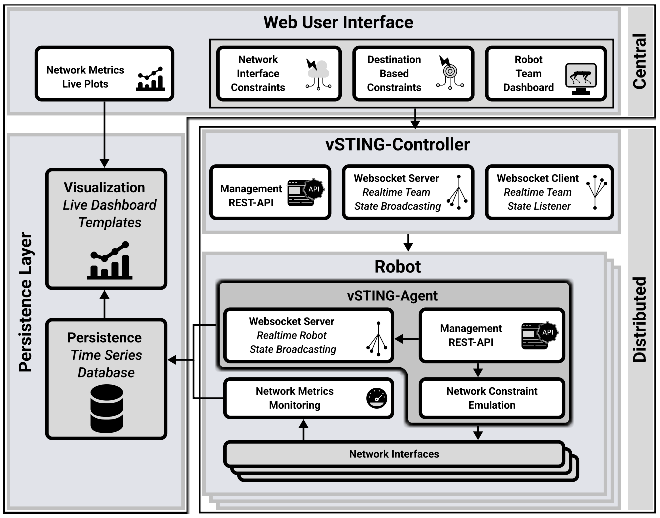

Fig. 5. The system architecture of the distributed vSTING solution. Core components such as the database, the user interface server and the vSTING-Controller are operated centrally while the vSTING-Agent is distributed to each robot.

is achieved for inbound traffic.

For outgoing traffic, the constraints for general interface shaping are applied on the *netEm* QDisc of the default branch.

IV. DISTRIBUTED vSTING IMPLEMENTATION

Building on the presented concept of remotely controlled distribued network emulation, this section explains the logical components of the distributed vSTING architecture: the vSTING-Agent, the persistence layer, the vSTING-Controller and the web user interface.

1) The vSTING-Agent: is the foundation of this architecture, since it exposes a REST-API for controlling network emulation control and enables network monitoring through the retrieval of network metrics and the broadcasting of network events via a websocket server.

2) The persistence layer: manages the persisting of the network metrics by storing them in a time series database. It also includes a visualization service that holds live dashboards and dashboard templates that can be embedded in web views. The visualisation service acts a frontend for the persistence layer and allows for a standalone use of this layer for network monitoring.

3) The vSTING-Controller: remotely coordinates the distributed network emulation by sending controls to the vSTING-Agent on each robot. It also listens to the websocket server present on each vSTING-Agent for network events to maintain an up-to-date representation of the network and connection status of each robot. Finally, it advertises the overall network status through a websocket server of its own.

4) The web user interface: provides a convenient network emulation control through the vSTING-Controller. By leveraging the live network status report of the latter, it can display a robot overview with connection status and, for each robot, the list of destination based and general interface constraints. Finally, it uses the dashboard templates provided by the visualization service of the persistence layer to create live graphs of the network metrics.

V. FUNCTIONAL EVALUATION AND TEST CASES

In this section, two test cases from application scenarios are formulated and evaluated to validate the features of the distributed vSTING.

A. *vSTING-Enabled Resource Management*

The first test case is inspired from the need for regulation during real-world multi-robot deployments. Applications running on the robots may unexpectedly exhibit increased network usage at a given time, taking bandwidth resources away from other robots. In this evaluation, we leverage the multi-robot support of the distributed vSTING to achieve resource management goals. The scenario features three network hosts, respectively representing a rover, a drone and an operator, all involved in a hypothetical rescue mission. Both robots are expected to report to the operator with a constant data stream. Furthermore, the drone additionally transmits another constant data stream to the rover. To validate the multi-robot support, different set of network constraints are applied in two phases. During the first phase, a general constraint is set to restrict the overall throughput of all robots. In the second constraint phase, in addition to the general constraint, a destination specific constraint is set to restrict the traffic from the drone to the rover. The datarates of the data streams between the network hosts and constraint details of the scenario are summarized in Tab.I.

TABLE I

PARAMETERS OF THE RESOURCE MANAGEMENT EVALUATION

	Parameter	**Value**
Streams	Rover to Operator	16 Mbps
	Drone to Operator	16 Mbps
	Drone to Rover	8 Mbps
Constraints	Phase Duration	30 s
	Phase 1	General Constraint
	Phase 2	Destination Based
	General Constraint	8 Mbps datarate limit
	Destination Based Constraint	4 Mbps datarate limit to Rover

The overview of traffic sent and received by each host during this evaluation is shown in Fig. 6. The general constraint performs as expected during the first phase. The overall output of each robot is restricted to 8 Mbps, which must be shared between the two active transmissions of the drone, so the throughput of the drone to the operator is lower than the rover's. In the second constraint phase, the drone stream to the rover is not affected to by the general constraint anymore, but by the destination based constraint instead. This increases the overall output of the drone and its throughput to the operator as well. Before the first phase, the difference in traffic being sent between drone and rover consists of the traffic sent by the drone the rover. It can

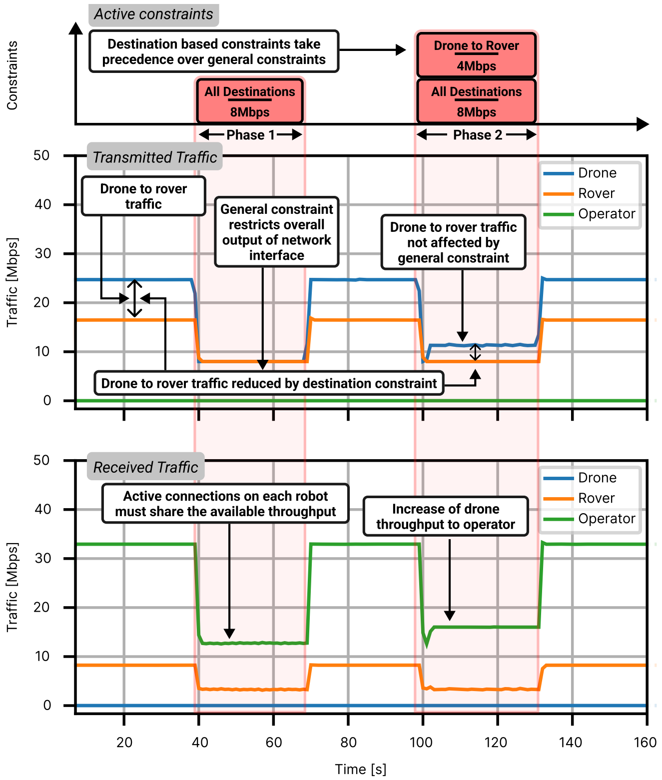

Fig. 6. Traffic sent and received by the network hosts while enforcing network constraints for resource management purposes. This illustrates the relationship between general and destination constraints.

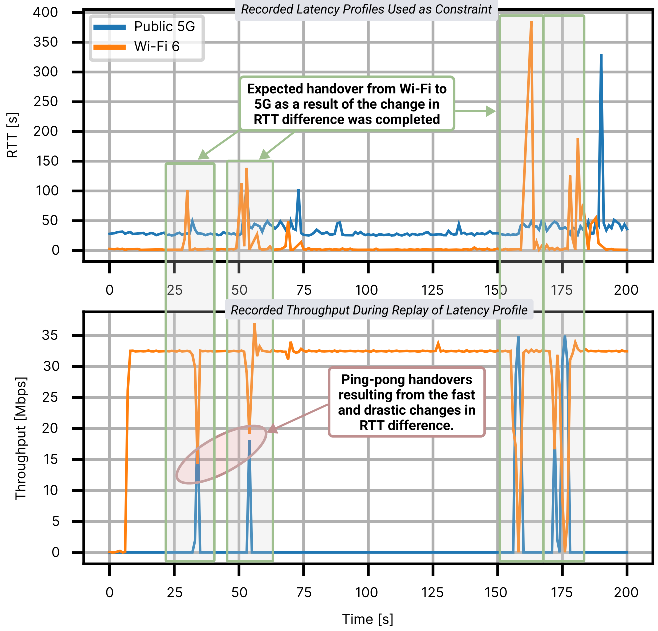

Fig. 7. Replayed RTTs of two network links to evaluate a multi-link scheduler during its development. The evaluated sheduler at this point of its development shows promising response to changes in the RTT difference.

be confirmed that this traffic is reduced during the second constraint phase. This experiment validates the multi-robot support of the distributed vSTING approach and presents it as a viable resource management solution.

B. *Network Replay Orchestration Powered Prototyping*

In the second test case, we use the coordinated network replay feature enabled by the distributed vSTING to prototype a simple multi-link scheduler of our own [14] in its early development stage . The purpose of the investigated multi-link scheduler's is to schedule traffic on the network link currently exhibiting the lowest Round Trip Time (RTT). The network replay orchestration of the distributed vSTING is used to verify the functionality of the scheduler and check its behavior. To this end, the RTT traces from a robot equipped with both Wi-Fi and 5G recorded during a previous mission are used as network constraints. These are shown in the upper part of Fig. 7 and are replayed by the distributed vSTING on the robot the scheduler is installed on. To visualize the scheduler's work, we generate test traffic of 30 Mbps.

The measured throughput of the robot using the multi-link scheduler is shown in the lower part of Fig. 7. It can be seen that at points in time where the RTT of the Wi-Fi link surpasses that of 5G, the scheduler starts sending the test traffic over 5G. However the short duration of these Wi-Fi RTT spikes causes the scheduler in this early version to initiate ping-pong handovers. A possible improvement could consist in observing the RTT for a longer period of time before making a handover decision. Another solution would be to improve the trace recording feature to allow a finer sampling of the RTT. Alternatively, it could also be possible to upsample the recorded traces, in order to broaden the spikes present in the recorded RTT profile. All of these options are future work goals, some already being undertaken and nearing completion.

VI. CONCLUSION

In this work, we presented the distributed vSTING, a software-only alternative to the previously proposed vSTING module. The distributed vSTING approach presented in this work enables the support of multi-robot and multi-link setups in evaluating the network behavior of robotic systems, especially teleoperated mobile ones, in degraded network environments.

To showcase the distributed vSTING functionality, we verified the multi-robot feature by using it simulatively as a resource management tool to limit the available bandwith for robots represented by network hosts. Next, we used network replay orchestration to verify the functionality of a multi-link scheduler in its early development version. These test cases validate the multi-robot and multi-link support of the distributed vSTING and highlight supported use-cases such as performing network resource management within a robot team. The impact of the distributed vSTING-Agent on the computing performance of the installation host is an important aspect we plan to investigate in detail in future work.

ACKNOWLEDGMENT

This work has been funded by the German Federal Ministry of Education and Research (BMBF) via the project DRZ (Establishment of the German Rescue Robotics Center, 13N16476) and was further supported by the project LARUS-PRO (14N15666), and the 6GEM research hub (16KISK038). We also thank Shashank Dixit for helping with implementation aspects of the vSTING-Controller and vSTING-Agent in the course of his student assistant activities.

REFERENCES

[1] F. Steinhäusler and H. V. Georgiou, “Detection of victims with uavs during wide area search and rescue operations,” in *2022 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2022, pp. 14–19.

[2] J. Cortes and M. Egerstedt, “Coordinated control of multi-robot systems: A survey,” *SICE Journal of Control, Measurement, and System Integration*, vol. 10, no. 6, pp. 495–503, 2017.

[3] A. Jacoff, H.-M. Huang, E. Messina, A. Virts, and A. Downs, “Comprehensive standard test suites for the performance evaluation of mobile robots,” in *Proceedings of the 10th Performance Metrics for Intelligent Systems Workshop*, ser. PerMIS ’10. New York, NY, USA: Association for Computing Machinery, 2010, p. 161–168.

[4] J. Pellenz, D. Dillenberger, and G. Steinbauer, “Novel rule set for the robocup rescue robot league,” in *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2016, pp. 282–283.

[5] M. Patchou, J. Tiemann, C. Arendt, S. Böcker, and C. Wietfeld, “Realtime wireless network emulation for evaluation of teleoperated mobile robots,” in *2022 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Sevilla, Spain, nov 2022, pp. 308–313.

[6] J. P. Queralta, J. Taipalmaa, B. Can Pullinen, V. K. Sarker, T. Nguyen Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund, “Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision,” *IEEE Access*, vol. 8, pp. 191 617–191 643, 2020.

[7] I. Majzik, O. Semeráth, C. Hajdu, K. Marussy, Z. Szatmári, Z. Micskei, A. Vörös, A. A. Babikian, and D. Varró, “Towards system-level testing with coverage guarantees for autonomous vehicles,” in *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS)*, 2019, pp. 89–94.

[8] M. Patchou, B. Sliwa, and C. Wietfeld, “Flying robots for safe and efficient parcel delivery within the COVID-19 pandemic,” in *2021 IEEE International Systems Conference (SysCon)*, Vancouver, Canada, apr 2021, pp. 1–7.

[9] M. Patchou, C. Arendt, P. Gorczak, J. Güldenring, J. Tiemann, and C. Wietfeld, “Hardware in the simulation loop framework for reproducible testing of rescue robot communications in constrained environments,” in *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Abu Dhabi, UAE, nov 2020.

[10] J. Güldenring, P. Gorczak, M. Patchou, C. Arendt, J. Tiemann, and C. Wietfeld, “SKATES: Interoperable multi-connectivity communication module for reliable search and rescue robot operation,” in *2020 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Thessaloniki, Greece, oct 2020.

[11] C. Arendt, M. Patchou, S. Böcker, J. Tiemann, and C. Wietfeld, “Pushing the limits: Resilience testing for mission-critical machine-type communication,” in *2021 IEEE 94th Vehicular Technology Conference (VTC-Fall)*, Virtual Event, sep 2021.

[12] C. Schüler, T. Gebauer, M. Patchou, and C. Wietfeld, “QoE evaluation of real-time remote operation with network constraints in a system-of-systems,” in *2022 Annual IEEE International Systems Conference (SysCon)*, Montreal, Canada, apr 2022.

[13] C. Krieger, H. Teper, J. Freytag, I. Priyanta, P. Schulte, M. Roidl, J.-J. Chen, and C. Wietfeld, “Integration of scaled real-world testbeds with digital twins for future ai-enabled 6g networks,” in *2023 IEEE Globecom Workshops (GC Wkshps), Workshop on Intelligent 6G Architecture: towards network simplicity and autonomy (Int6GArch)*, Kuala Lumpur, Malaysia, dec 2023.

[14] T. Gebauer, M. Patchou, and C. Wietfeld, “Seamless: Radio metric aware multi-link transmission for resilient rescue robotics,” in *2023 IEEE International Conference on Safety, Security, and Rescue Robotics (SSRR)*, Fukushima, Japan, nov 2023.