Accepted for presentation in: 2022 IEEE International Symposium on Safety, Security, and Rescue Robotics

Realtime Wireless Network Emulation for Evaluation of Teleoperated Mobile Robots

Manuel Patchou[1], Janis Tiemann[1], Christian Arendt[1], Stefan Böcker[1] and Christian Wietfeld[1]

***Abstract*— The evaluation of remotely supervised robotic systems must include exposure to probable network disturbances to assess and tune their behavior for similar situations. This paper presents the vSTING module, a solution to evaluate a system's network resilience with minimal installation overhead. It can subject a network link to constraints provided in various ways: user input, recorded network traces, or location-based. In the first evaluation step, the general ability to constrain a network link is confirmed. Next, the teleoperation performance of multiple robots challenged. Finally, the challenging network environment recorded during a mission is replayed. The evaluation validates vSTING as a useful tool to assess and develop the resilience of systems against the network disturbances expected with real-world wireless connectivity.**

I. Introduction

Imbuing robots with practical features such as enhanced mobility ranges and freedom of movement introduces challenges for reliable connectivity. Robotic applications are expected to cope with connectivity problems that may arise on the field during missions. Therefore, evaluations of their resiliency and robustness must be driving factors of their development process.

While various tools for simulating network disturbances already exist for personal computers, there is still untapped potential and significant interest in solutions that target robotic use cases and account for robotic's specificities. This paper presents a physical solution, the virtual STING (vSTING) module, that introduces network constraints in existing network infrastructures with minimal overhead to evaluate and develop network resilient applications. It extends on the work presented in [1], where networks are stress-tested with a spatially distributed traffic and interference generator, as it provides portable alternative to conduct network resiliency evaluations. The contributions of this work can be summarized as depcited in Fig. **??**with the operation modes supported by vSTING:

- **Manual activation** of user-defined network constraints
- **Predefined standard** constraints
- **Recorded traces based** constraint activation
- **Location-based** constraint activation.

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

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

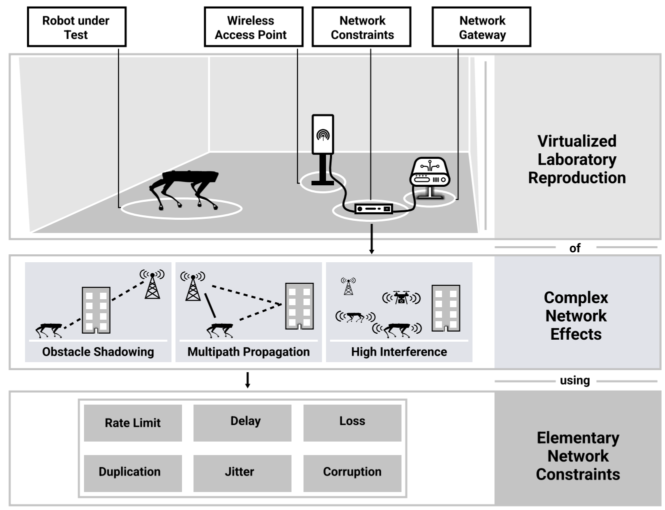


Fig. 1. The envisioned approach for evaluating robotic applications under realistic connectivity conditions: leveraging simple network constraints through network emulation to rebuild complex network effects on live traffic.

II. Related Work

The National Institute of Standards and Technology (NIST) has included radio communications tests [2], [3] in its published suite of standard test methods for evaluating emergency response robot capabilities in repeatable ways [4]. This stresses the importance of assessing network-reliant robotic systems and applications in degraded network environments. A compilation of such applications is provided in [5]. Furthermore, This assessment must be application-aware and based on real-world scenarios to adequately supplement the isolated unit tests of components [6].

Assessing the network capabilities of a system in a targeted environment can be handled in gradual steps. Simulations may help preview possible communication performance in the targeted deployment environment. A joint simulator for network and mobility [7] was specialized for handling hybrid fleets of aerial and ground vehicles to cover use cases of drone-enabled parcel delivery [8] [9]. The impact of the communication channel on the information sharing between Unmanned Aerial Vehicles (UAVs) is investigated in [10]. In [11], a network simulation is conducted with Network Simulator version 3 (ns-3) to observe the performance of the second version of the Robotic Operating System (ROS) in lossy network environments.

Hardware in the Simulation Loop (HiSL) simulations can strengthen the validation of simulation results. In [12], a

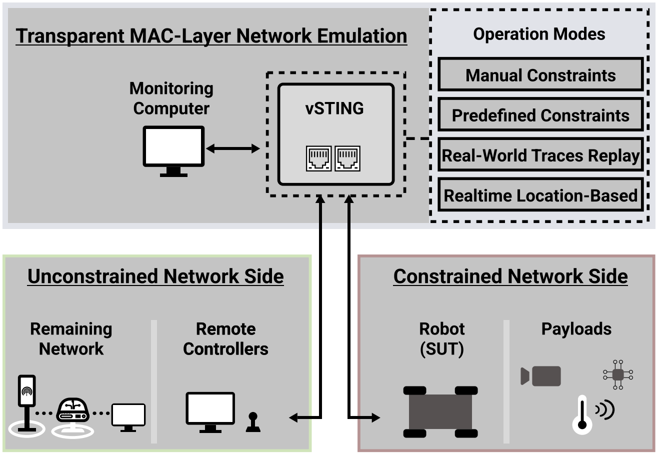


Fig. 2. Concept for the encapsulation of network constraints in a module: Network emulation applied on transparently forwarded traffic by the module.

framework to build HiSL simulations was used to confirm the multi-Radio Access Technology (RAT) features of a novel communication module [13].
Further validation of simulation results can be obtained through experiments with real entities. To observe the resilience of systems in networks with interfering traffic, a solution using spatially distributed real nodes to generate traffic was proposed in [1].
Due to costs and logistics, experimenting with real entities is not always possible. Simulating or emulating specific aspects within a real entity is a feasible compromise to speed-up iterative development processes. As asuch, network emulation was used in [14] to cost-effectively evaluate the impact of usual network constraints such as latency and packet loss on real-time teleoperated driving.

III. Network Constraints As a Module Approach

The concept of the vSTING module can be summarized as follows: emulate network constraints on forwarded traffic. The strong point of this approach is that the network infrastructure remains unchanged, as the module merely is inserted between operator and robot, thus dividing the network into two sides, as illustrated in Fig. 2. The vSTING module acts as a switch between the constrained and the unconstrained side, so the content of the forwarded traffic is unchanged. Traffic forwarding and network emulation were chosen as the core ideas of this concept, for their benefits regarding ease of installation and modularity of the resulting solution.

A. *Transparent Traffic Forwarding*

The nature of the traffic forwarding was fixed while considering the layers of the Open Systems Interconnection (OSI) model and their roles. The layers above layer three (network) are not suitable as they involve complex aspects like routing that increase installation overhead. For example, the network layer uses the Internet Protocol (IP) and packets are sent to a specific next-hop IP address based on the destination's IP address. Forwarding traffic on this layer, therefore requires knowledge about the network's IP configuration.
Forwarding on traffic on the layer two (data link) however does not require any configuration as it uses the physical addresses (Medium Access Control (MAC) addresses) of the network devices. The vSTING forwards on layer two by using a network bridge to unite the network segments connected to its network ports, thus behaving like a switch.

B. *Network Emulation*

To induce network constraints, network emulation is performed on the network interfaces bound to the network bridge built for transparent traffic forwarding as explained in III-A. The Linux Operating System (OS) provides tools for networking and traffic control, which is the set of queueing systems and mechanisms governing how network packets are sent and received.
Among these tools, we rely mainly on *Tc*, a tool designed for configuring traffic control by using Queueing Disciplines (QDiscs). QDisc is the abstraction used by the Linux kernel at the data link layer to schedule outgoing packets of a network device on the physical layer. *NetEm* is a QDisc that specializes in network emulation by giving control over parameters such as delay, jitter, datarate, packet loss, corruption, and duplication. The specified parameters are applied to the packet queue of the selected network device.

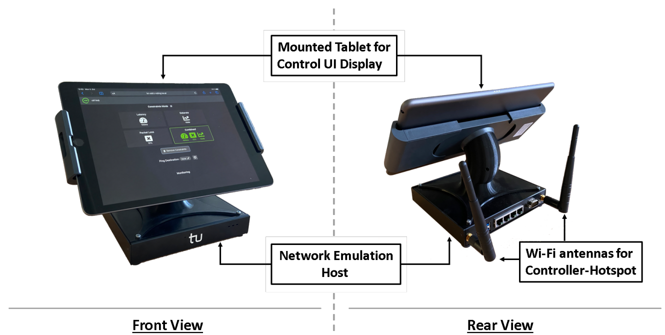


Fig. 3. Physical realization of the vSTING module, comprised of an embedded computer for network emulation and a mounted tablet to display the UI controls.

IV. Implementation of the vSTING Module

Building on the presented concept of network emulation on forwarded traffic, this section elaborates on the material and logical components of vSTING.

A. *Hardware Platform*

Fig 3 shows the physical realization of the vSTING solution, which entails a single board computer with a 1GHz Quad-Core 64 Bit processor and 4GB working memory as the network simulation and forwarding host, and a mounted tablet for displaying the UI for the vSTING control. The network emulation host opens a Wi-Fi hotspot over which the tablet connects to access the emulation's control UI. The deployment of vSTING in a network between two hosts or network sections merely requires plugging-in two RJ45 cables so that the vSTING sits between them as a switch.

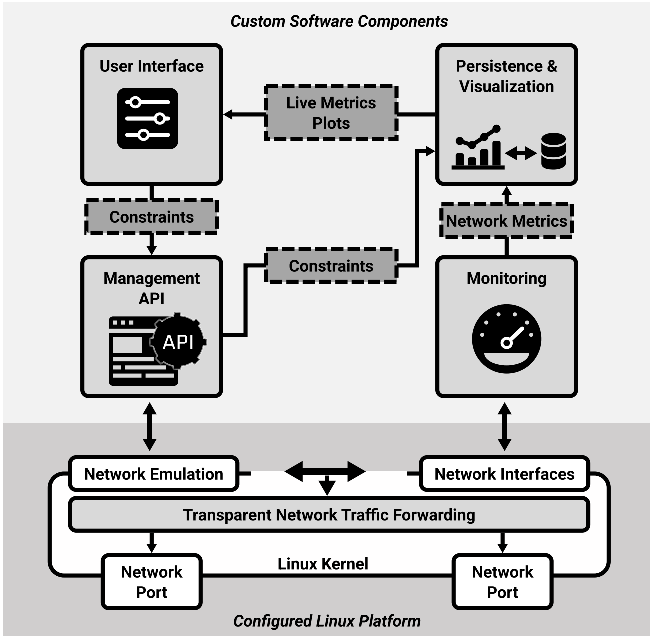


Fig. 4. Logical architecture running on the hardware of the vSTING module. The developed additions run atop the Linux kernel's network emulation applied on the forwarded network traffic.

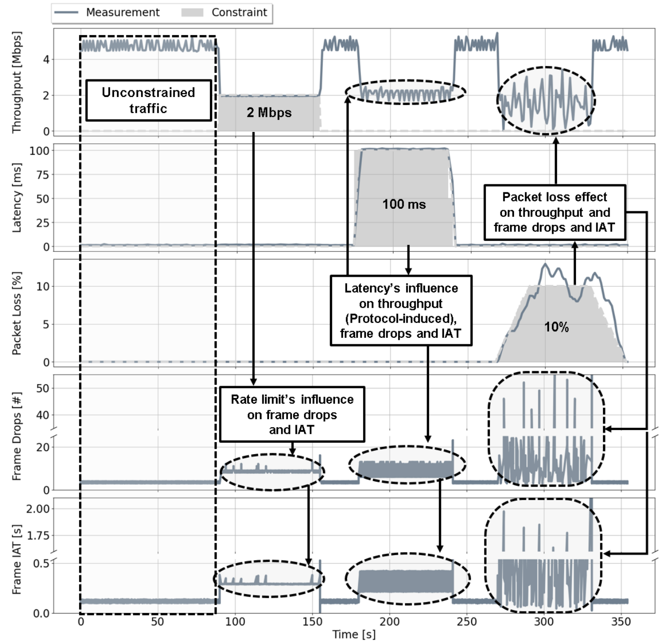


Fig. 5. Manual enforcement of constraints on live traffic of a robot's FPV Camera. The physical dependencies between the network metrics and protocol-induced cross-metrics effects can be observed. Frames drops and Inter Arrival Time (IAT) also monitored, to capture the constraints' effect on the FPV stream.

B. Software Components

The logical architecture of the vSTING illustrated in Fig 4 features custom software components running on a preconfigured Linux system. The software components' purpose is the network emulation control and the recording, storage, and display of network metrics.

1) Metrics Monitoring:

To monitor the impact of the network emulation both active and passive measurement methods are used.
Passive measurement methods are based on observations of an undisturbed and unmodified packet stream of interest and depend on the existence of one or more packet streams to supply measurements. Active methods inversely generate the packet streams of interest for the measurement [15].
The incoming and outgoing data rates are measured passively, while the delay and packet loss are measured actively, by generating an additional minimal stream of Internet Control Message Protocol (ICMP) packets. When monitoring network interfaces associated with wireless networks, signal strength information is also collected.

2) Persistence and Visualization:

The collected network metrics are stored in a time series database. For an immediate overview of the network behavior, data visualization software is used to define dedicated dashboards for rendering specific metrics.
The main dashboard for vSTING offers a live view of basic network metrics: delay in milliseconds, packet loss in %, and incoming and outgoing traffic in bytes per second. Network constraints are visualized as reference value for the corresponding metric.

3) Control User Interface:

The dashboards created in the data visualization software are integrated into the control UI of the vSTING and provide immediate feedback on network constraints updates. The control UI is a frontend web app hosted on the network emulation host. It presents the user with the predefined network constraints bundles such as 100 ms additional latency or 10% packet loss. An expert mode that allows specifying custom constraints is also available.

4) Management Application Programming Interface:

A backend server runs on the network emulation host and exposes an Application Programming Interface (API) to manage the network emulation. The management API converts the constraints it receives in *NetEm* QDiscs for *Tc* and applies them based on the selected operation mode. The constraints may either originate from the Control UI through manual user input, from previously recorded and uploaded network traces, or from a Radio Environmental Map (REM) with the current robot's location as a cursor. Finally, the constraints applied are forwarded to the data visualization tool to represent them alongside the collected network metrics as reference values for visual validation.

V. Functional Evaluation and Test Cases

In this section, the artificial network degradation functionality proposed by the vSTING module and implemented by following the network constraint as a module approach is evaluated. The evaluation features three use-case scenarios.

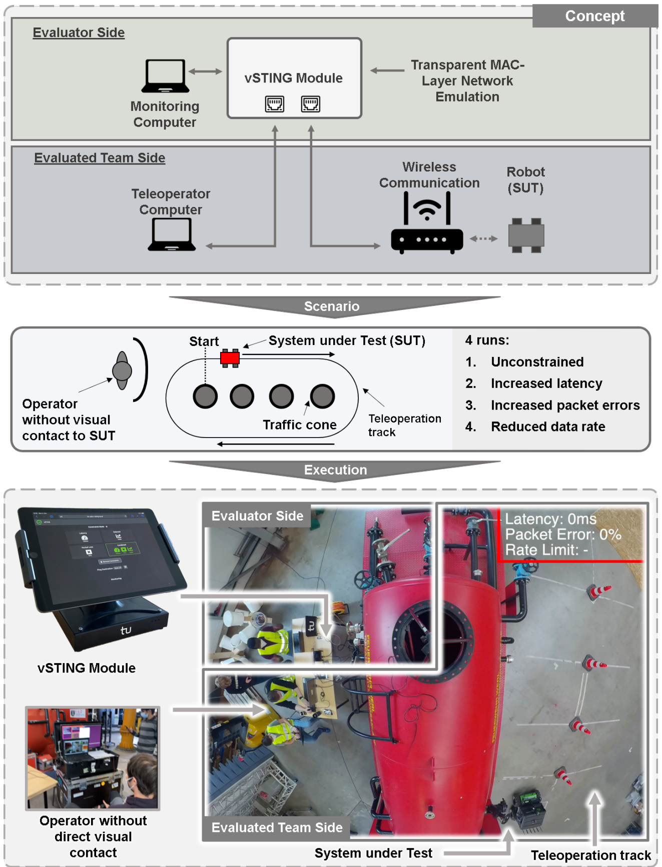
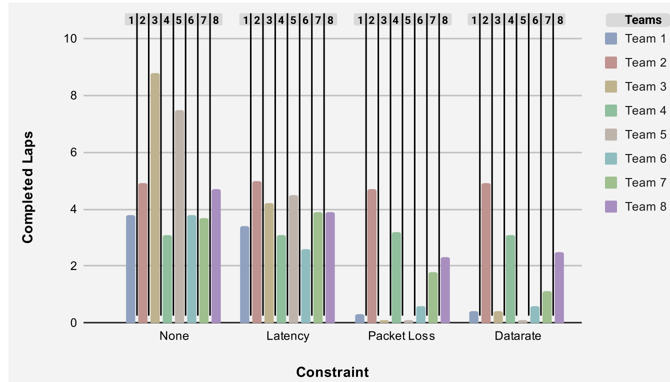


Fig. 6. Setup for evaluating the teleoperation performance under network constraints. The robot operator must complete as many laps as possible on the defined course during multiple three minutes runs with different vSTING-enabled network constraints.

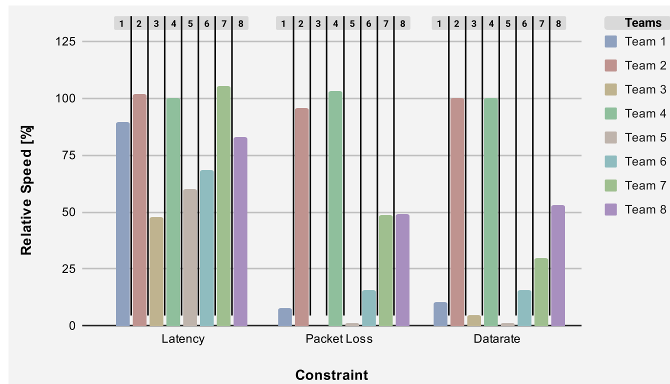
A. Manual Constraint Application

In the first evaluation scenario, specific constraints are manually activated and disabled on a network link. The constraints considered in this evaluation are 2 Mbps data rate limit, 100 ms latency and 10 % packet loss. This evaluation scenario is conducted with an exploration robot that transmits an FPV stream over the network, thereby generating ca. 5 Mbps network traffic using a transmission scheme based on the Transmission Control Protocol (TCP) protocol. This scheme allows the transmission of a data packet only after receipt of the preceding one is acknowledged. Therefore, a strong dependency between the transmitted traffic, and the latency and packet loss is expected.

The constraints are applied separately and consecutively as shown in Fig 5, where the network recorded metrics are visualized. When the traffic is unconstrained, a frame rate of 10 Hz can be observed for the FPV stream since the Inter Arrival Time (IAT) lies around 100 ms.
The data rate constraint of 2 Mbps is applied first and halves the overall measured traffic. While the frame drops are doubled, the frame IAT increases to 200 ms, thus halving the FPV stream's frame rate to 5 Hz.



(a) Absolute Teleoperation Performance.



(b) Teleoperation Performance relative to unconstrained run.

Fig. 7. Results of the teleoperation evaluation with vSTING-enabled network constraints.

After disabling the data rate limit, a delay of 100ms is added with the latency constraint's activation. This nearly also halves the throughput as the increase in frame drops and IATs shows. The increased latency leads to a longer transmission time for each frame. This raises the number of dropped frames and reduces the frame rate. On average, the frame rate is also halved and shows greater variation. The variation is modulated by the frame drops, as for each dropped frame, the IAT increases, and the frame rate is reduced. The packet loss constraint is applied last, after disabling the latency constraint. The measured packet loss increases accordingly and a decrease in the transmitted traffic is observed. The statistical nature of this constraint causes even greater variations in the frame drops and IATs.
The network metrics recorded during this evaluation confirm the network degradation functionality of the vSTING module and highlight the protocol-induced effects which may arise in a different metric while applying a network constraint.

B. User Centered Evaluation in Robotic Context

Having confirmed the function of the vSTING with the manual constraint evaluation, an evaluation from the user's perspective is carried out next to showcase a vSTING deployment to challenge and assess an actual robotic application: mobile robot teleoperation.

Starting from the vSTING concept presented in III, we

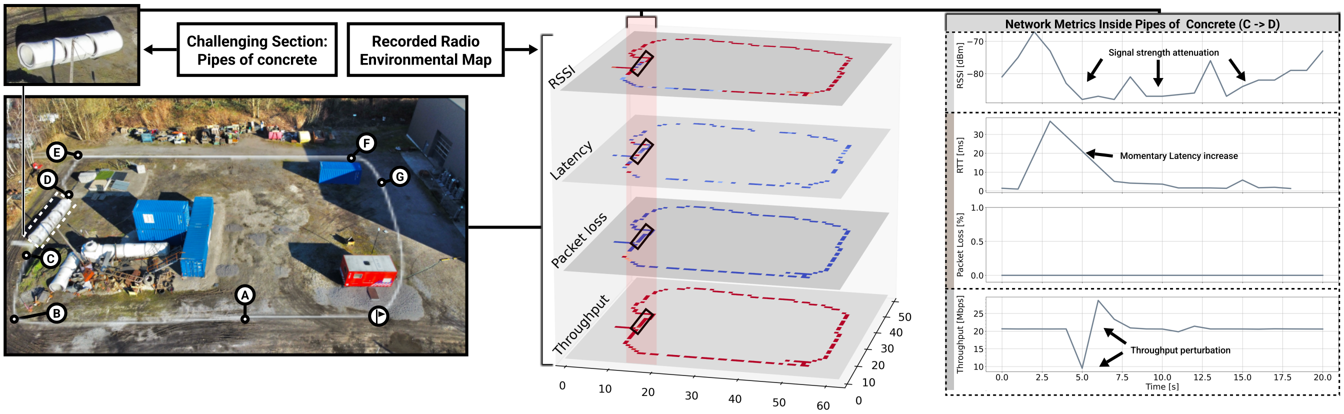


Fig. 8. Scenario setup for the network replay evaluation: The outdoor test field scenario of the German Rescue and Robotics Center furnished with Wi-Fi connectivity. The obstacles present in the field yield effects on the observed connectivity, especially when the robot moves through the pipe system at the left edge. This specific part of the metrics was taken for evaluating the network replay functionality.

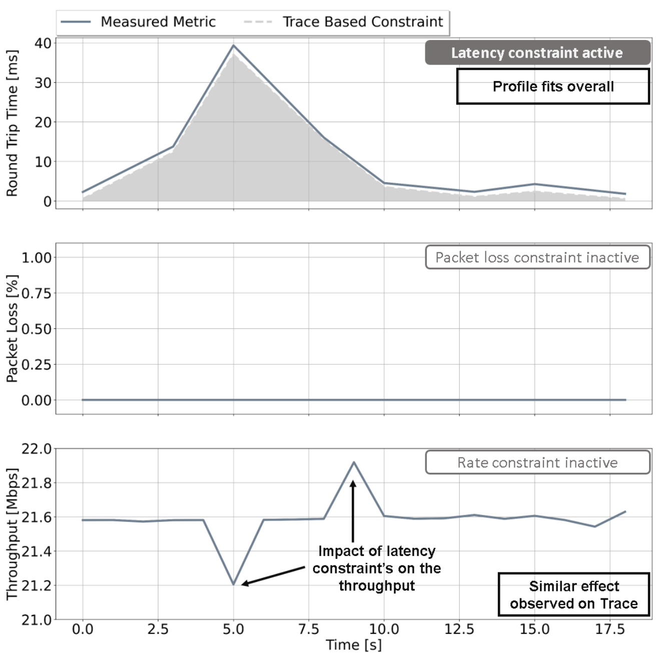


Fig. 9. Network replay while using only the latency traces as constraints. The generated metrics for the latency match the recorded trace profile. The throughput shows a similar trend to the traces in response to latency fluctuations.

formulate a simple evaluation scenario, as illustrated in Fig. 6, where the teleoperation performance can be measured under vSTING-enabled network constraints. In each run of the formulated scenario the human operator must lead the robot to complete as many laps as possible on a simple loop-shaped track in the given time of three minutes while having no direct visual contact with the robot. The robot's sensors and video feed therefore are the only source of thruth to navigate the robot. Each run exhibits a different network environment. The first run is conducted without any network constraints, while the subsequent ones feature respectively:

- 100 ms additional latency
- 10 % packet loss
- 10 Mbps data rate limit

Six international teams with eight robots overall took part in this evaluation, with their own meticulously assembled robotic systems, thereby using different network approaches. The specifics of the robotic setups and their impact on the observed performance results presented in Fig. 7 are part our future work subjects.

Fig. 7a presents the absolute score of each participating robot in the unconstrained and constrained runs expressed as the number of completed laps. The impact of the network constraints is stressed in Fig.. 7b, where the number of completed laps is expressed as a percentage of the number of laps completed in the run without network constraints.

A noticeable fact is a relatively constant performance of team 2 and team 4. Team 4 had their robot navigating autonomously after using the run without constraints to set waypoints. Team 2, on their end, used a video server with adaptive network transmission capabilities that preserved the teleoperation capacity through all the network constraints by adjusting the data rate. While most of the teams were still able to operate under an additional latency of 100 ms, the packet loss and data rate constraints proved to be more of a challenge and most teams had to reduce the traffic produced by the different camera systems and sensors installed on their robots to retain teleoperation capability.

The overall results indicate great potential in the resilient design and architecture of robotic network communications. The impact of the network constraints varies significantly depending on the constraint's nature and is mostly defined by the networking solution in use on each robot.

C. Network Replay: Trace-Based Constraint Application

This evaluation showcases the ability to apply constraints based on recorded traces of real missions. In this case, a test exploration mission was performed on the outdoor test field of the german center for rescue robotics (Deutsche

Rettungsrobotik-Zentrum (DRZ)), shown on the left in Fig 8 in Dortmund. One challenging section of this mission path is the segment C $\rightarrow$ D which goes through pipes made of concrete. Strong radio signal attenuation and network experience degradation inside the pipes are expected and confirmed with a look at the network metrics, taken from the REM corresponding to that path segment and shown on the right side of Fig. 8. For a clearer observation of the network effects without protocol influence, the User Datagram Protocol (UDP) protocol was used for data transmission.

vSTING was used to replay the Round Trip Time (RTT) from the recorded network traces as latency constraint to reproduce the observed network behavior at a later point in time, once we were away from the test field. The result is shown in Fig. 9. The generated latency profile follows the applied constraints and matches the RTT trace. The momentary increase in latency observed in the traces, which was caused by the sudden decrease in signal strength could be reproduced.
Furthermore, the throughput fluctuation as an effect of the latency increase resulting from the pipes' signal attenuation manifested as well. The fluctuation features a small decrease followed by an increase of equal scale caused by the jitter resulting from the latency variation. With the latency increase, some packets in transmission are delivered later, therefore causing the throughput reduction. With the latency decrease, some of the packets being transmitted are delivered sooner and momentarily raise the throughput.
This evaluation highlights the hidden impact of jitter as a network constraint. Due to the natural or protocol-induced interactions between network metrics, replaying more than one metric at a time may be a challenging task and is also one of our future work subjects.

VI. Conclusion

In this work, we presented vSTING, a compact solution to evaluate the network behavior of robotic systems, especially teleoperated mobile ones, in degraded network environments. To showcase the vSTING functionality, we manually applied constraints on the FPV stream of an exploration robot, while monitoring the network metrics. That confirmed its ability to enforce network constraints. Next, vSTING was used in a challenge with eight different robots featuring different network setups to measure the impact of network constraints on teleoperation performance. Finally, network traces of a challenging environment were successfully used as constraints to reenact the observed network behavior.
The teleoperation challenge showed there is still great potential in resilient communications design for robotics. To contribute tools for this goal we plan to address the replay of multiple network metrics and a deeper analysis of the results and network architectures of the robots that took part in the teleoperation challenge in our future work.

Acknowledgment

This work has been supported by the German Federal Ministry of Education and Research (BMBF) in the project A-DRZ (Establishment of the German Rescue Robotics Center, 13N14857). Regarding the teleoperation evaluation: we thank the chairs of the Robocup Rescue German Open 2021, Stephan May and Johannes Pellenz, for coordinating the evaluation as a part of the competition. We thank Kevin Daun and Marius Schnaubelt of the TU Darmstadt for facilitating the necessary preparatory work and communication with the teams. We thank the participating teams: AutonOhm, Capra, Dynamics, Francor, Hector and Siemens Space Agency. We thank the DRZ - German Rescue Robotics Center, for the allowed usage of their facilities for the teleoperation and network replay evaluations.

References

[1] 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.
[2] "ASTM E2854-12, Standard Test Method for Evaluating Emergency Response Robot Capabilities: Radio Communication: Line-of-Sight Range," ASTM International, Tech. Rep.
[3] "ASTM E2855-12, Standard Test Method for Evaluating Emergency Response Robot Capabilities: Radio Communication: Non-Line-of-Sight Range," ASTM International, Tech. Rep.
[4] 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.
[5] 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.
[6] 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.
[7] B. Sliwa, M. Patchou, and C. Wietfeld, "Lightweight simulation of hybrid aerial- and ground-based vehicular communication networks," in *2019 IEEE 90th Vehicular Technology Conference (VTC-Fall)*, Honolulu, Hawaii, USA, sep 2019.
[8] M. Patchou, B. Sliwa, and C. Wietfeld, "Unmanned aerial vehicles in logistics: Efficiency gains and communication performance of hybrid combinations of ground and aerial vehicles," in *IEEE Vehicular Networking Conference (VNC) 2019*, Los Angeles, USA, dec 2019.
[9] ——, "Flying robots for safe and efficient parcel delivery within the COVID-19 pandemic," in *2021 IEEE 15th International Systems Conference*, Vancouver, Canada, apr 2021.
[10] S. Moon, J. J. Bird, S. Borenstein, and E. W. Frew, "A Gazebo/ROS-based communication-realistic simulator for networked sUAS," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, Athens, Greece, 2020, pp. 1819–1827.
[11] P. Thulasiraman, Z. Chen, B. Allen, and B. Bingham, "Evaluation of the robot operating system 2 in lossy unmanned networks," in *2020 IEEE International Systems Conference (SysCon)*, 2020, pp. 1–8.
[12] 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 *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Abu Dhabi, UAE, nov 2020.
[13] 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.
[14] 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.
[15] A. Morton, "Active and passive metrics and methods (with hybrid types in-between)," Internet Requests for Comments, Internet Engineering Task Force (IETF), RFC 7799, may 2016.