SKATES: Interoperable Multi-Connectivity Communication Module for Reliable Search and Rescue Robot Operation

Johannes Güldenring, Philipp Gorczak, Manuel Patchou, Christian Arendt, Janis Tiemann, Christian Wietfeld
Communication Networks Institute (CNI), TU Dortmund University, Dortmund, Germany
{ johannes.gueldenring, philipp.gorczak, manuel.mbankeu, christian.arendt janis.tiemann, christian.wietfeld }@tu-dortmund.de

***Abstract*—Robots are used more and more often in search and rescue scenarios, for example in post-disaster situations, which are dangerous or lethal for humans, when searching for missing persons after earthquakes. Here, reliable communication is indispensable: on the one hand, unmanned vehicles need to be teleoperated in real-time. On the other hand, high-resolution images, video and sensor-readings producing huge data volume must be transported from the robot back to the operator or rescue forces. This work presents and evaluates SKATES, a new solution enabling interoperable multi-connectivity for reliable communication between a search and rescue (SAR) robot and its remote operator. In this paper, the SKATES concept, its implementation and subsequent validation in a field experiment are presented. SKATES operates protocol agnostic and tunnels any data from the robot reliably to the operator using an improved SOCKS proxy. To increase robustness, a Multi-Radio-Access-Technology (Multi-RAT) approach incorporating cellular as well as WiFi connectivity options is implemented. This enables link aggregation as well as smooth handover between different wireless technologies in case of outages of one technology. As part of the experimental validation, an unmanned ground vehicle (UGV) has been equipped with a 360° 8K resolution camera, a First Person View (FPV)-camera and the proposed SKATES solution. Being teleoperated, the robot explores a building, while streaming live camera data to the remote operator. The experimental results underline the feasibility and good performance of the approach in a challenging environment: when single communication links break down or do not provide sufficient capacity, the SKATES-enabled Multi-RAT approach enables smooth handovers between different RATs and ensures reliable and real-time communication during the whole experiment.**

***Index Terms*—rescue robotics, search and rescue, SAR, Multi-RAT, multi-link, multipath, shadowsocks**

I. Introduction

Recent advances in robotics and communication technology are bringing new unmanned systems applications within reach. Among these applications, the use of robotic systems in post-disaster scenarios has gathered interest due to the challenging but vital nature of disaster response. However, this use-case poses requirements beyond robustness, speed and versatility [1] of robot platforms: Unless an unmanned system is able to explore, navigate, perform its task and return fully autonomously, a reliable communication link between the robot and an operator has to be available at all times. To allow the operator to get an overview of the situation and make appropriate decisions, a variety of information such as live telemetry, video streams, and sensor measurements need to be relayed between the robot and its control station.

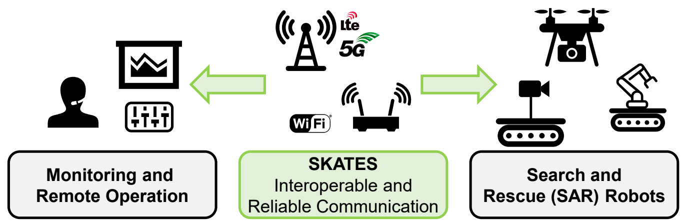

Fig. 1. Schematic illustration of the proposed communication solution: The interoperable communication module is attached to the SAR robot. The robot's proprietary protocols are made more robust due to the multi-connectivity of the modules and it is now able to be reliably remotely operated

Today, two wireless technologies are widely available and able to deliver the high bandwidth and low latency required for remote SAR robot control. Among cellular communications, the roll-out of the fifth generation system (5G) has just started while its predecessor LTE is widely deployed and well understood. On the local area network side, the IEEE 802.11 standards are widely used with the latest generation WiFi-6 (IEEE 802.11ax) facing an increasing degree of adoption. Both technologies offer high data rates and low latency communication but have unique advantages and drawbacks in terms of infrastructure deployment and available spectrum. Therefore, this work proposes SKATES[1], an interoperable multi-connectivity communication concept, which enables reliable communication between operators and unmanned systems. The approach is transparent and protocol agnostic which makes it backwards-compatible to many existing applications and systems. To maximize reliability, SKATES implements a multi-connectivity Multi Radio Access Technology (Multi-RAT) approach, where the communication module is equipped with multiple modems and aggregates available network resources, while presenting a transparent interface to applications on the robot and the control station. This allows seamless handovers between different RATs when connection quality varies or the signal drops without requiring a change in application logic.

[1]The name SKATES originates from the used SOCKS protocol, which is empowered to run smoothly and reliably in a highly mobile scenario.

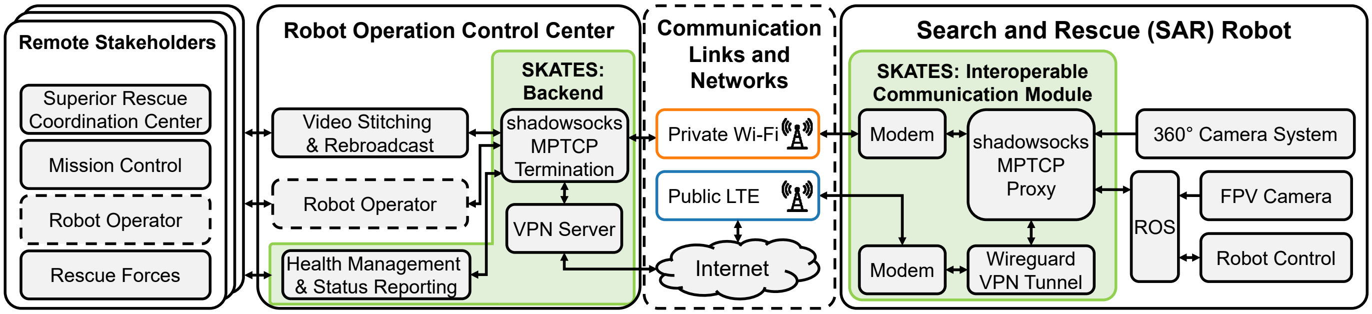

Fig. 2. Detailed architecture of the proposed SKATES concept. SKATES transports data between the SAR robot and the remote operator. It operates protocol agnostic in a transparent way by using a shadowsocks MPTCP proxy to increase reliability by exploiting multiple communication link. The illustration uses the same Wi-Fi + LTE configuration, which is also used in the experimental validation. However, the concept is not limited to two RAT, alternate configurations, e.g. using multiple LTE interfaces, are also feasible.

The SKATES implementation uses the Transmission Control Protocol (TCP) extension Multipath TCP (MPTCP) to aggregate multiple communication links. Links traversing the public Internet, such as links using public cellular operators, are tunneled through a Virtual Private Network (VPN) to traverse Network Address Translation (NAT) as well as firewalls and middle-boxes, which may block protocol extensions [2]. In addition, we propose a modified and enhanced shadowsocks [3] to enable non-MPTCP applications to make use of the Multi-RAT approach and benefit from smooth RAT handovers.

To prove the feasibility and advantages of the approach, the implementation is evaluated experimentally. We set up a challenging indoor environment using the Husky robot platform. The robot is equipped with a 360° 8K high resolution camera as well as an First Person View (FPV) camera, and the SKATES module, using a LTE and one Wi-Fi communication configuration. The module adaptively routes traffic through both networks to maintain a stable live video stream while allowing responsive remote control of the robot at all times. A video of the experiments can be found online at [4].

II. Related Work

The underlying work picks up the idea of a Multi-RAT approach based on MPTCP [5]. First experimental studies of the author have proven the increase of reliability and throughput of the Multi-RAT concept both analytically [6] as well as an in experiments [7]. The previous work has been conducted in a maritime SAR scenario, where the communication channel is more predictable due to the line-of-sight than in the setup of this work. Next to MPTCP the Multipath QUIC (MPQUIC) protocol [8] can be used to achieve a similar link-aggregation. However, MPQUIC lacks maturity and due to running in user space instead of kernel space a performance degradation may be expectable. Therefore, the underlying work makes use of MPTCP. An MPTCP specialization for video streaming is Multipath Dynamic Adaptive Streaming over HTTP (MP-DASH) [9]. MP-DASH splits the video into chunks and defines deadlines for data transmissions. However, the paper discusses latencies of eight to ten seconds, which is too high for real-time video streaming. A recently published real-time video streaming approach is discussed in [10]. The authors propose a fast concurrent transfer to minimize the Round Trip Time (RTT) for video frames. However, this approach is not feasible in the underlying study, as an live inspection of the video stream was not possible because of the minimal communication processor footprint, which is unable to re-encode the video, and proprietary interfaces of the 360° camera system. The application of MPTCP for robotic systems has been proposed in [11] where control and data links are managed via a specifically developed scheduler. The proposed algorithm is able to fail-over traffic flows between the two links while ensuring that control traffic always has the highest priority. Another approach for increasing reliability is to adapt transport layer protocols for operation in unreliable wireless networks as proposed by [12]. While mechanisms on the transport layer can improve reliability of critical traffic flows through prioritization or similar measures in situations of degraded link quality, this work evaluates the aggregation of multiple links to maintain a constant high bandwidth connection.

III. System Concept and Implementation

The core idea of the proposed SKATES concept is a Multi Radio Access Technology (Multi-RAT) approach that maximizes communication reliability, while being highly compatible due to transparent handling of application protocols. Figure 2 shows the full architecture diagram of SKATES and its interacting components. The figure is structured into topological segments. The SKATES implementation (highlighted in green) consists of an interoperable communication module attached to the robot, and the backend located in the robot control station. The data flow between processes on the robot and control station interfacing with SKATES is transparent. The communication module and the backend are connected through multiple links using different RATs.

The example robot system shown in the architecture diagram in Figure 2 is equipped with two cameras for SAR use cases. While the robot controller and one of the cameras have Robot Operating System (ROS) interfaces, the second camera exposes a proprietary TCP endpoint. At the control station,

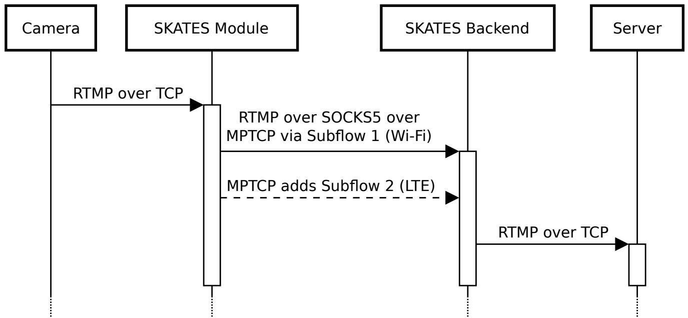

Fig. 3. Illustration of the connection establishment at the example of one video stream (works in the same way for the ROS-, FPV or any other data are handled in the same way). The MPTCP shadowsocks proxy enables transparent Multi-RAT for the proprietary camera system.

Fig. 4. The communication dashboard of the health monitoring in the SKATES backend shows the key-performance indicators throughput, latency and signal quality for the individual interfaces as well as for the combined multi-link.

the camera streams are reassembled for presentation. All information are re-broadcast and published to off-site rescue coordination, mission control and rescue forces. Therefore, the robot operator can work remotely as well as on-site with access to the video streams and health and status monitoring. This example configuration is also used in the experimental validation presented in IV. Additional sensors or proprietary applications can be added with minimal integration effort as the SKATES implementation provides a transparent tunnel.

A. Transparent Protocol Tunneling

In order to maximize interoperability with existing components and protocols, the SKATES backend and modules include a proxy server and clients. All data traffic from the robot is redirected to the proxy process using local routing rules. We utilize a SOCKS implementation, which creates a tunnel on Open Systems Interconnection (OSI)-model session layer ("Layer 5") and is therefore transparent for applications. The SOCKS protocol, first approved by Internet Engineering Task Force (IETF) in 1996, has been shown to achieve good performance with little runtime overhead even on mobile devices [13]. Within the scope of this work the SOCKS proxy implementation shadowsocks [3] is used, based on the previous work of Coninck et al. [14].

Data flows between ROS processes are forwarded between the two ROS instances on the robot and the control station using a specialized gateway implementation that has been developed for rescue robotics applications [15]. The process transports a subset of data topics between the robot and control station and is able to apply quality of service measures such as rate limiting. All topics are multiplexed via a single TCP connection which in turn uses the SOCKS tunnel described above.

Figure 3 shows the proxied protocol connection establishment for an Real-Time Messaging Protocol (RTMP) camera stream. Instead of establishing a direct connection to the backend, the handshake is redirected to the shadowsocks proxy on the SKATES module. Here, the process creates an outgoing connection to the proxy server. The server itself then creates the final connection to the destination RTMP endpoint in the backend. The multi-link capability, between the proxy endpoints, which is also shown in the Figure, is illustrated in the following section.

B. Enabling Multi-Connectivity and Seamless Handovers

Aggregating multiple communication technologies to maximize reliability is one of the proposed approach's key ideas. Within the scope of this work, the MPTCP Linux implementation is utilized for distributing traffic over multiple redundant Internet Protocol (IP) routes provided by the communication links. MPTCP is able to switch between different interfaces and links and thus allows seamless handovers between RATs such as LTE and Wi-Fi. As robot systems and payload modules may be closed and proprietary, the installation of MPTCP into the network stacks on all components is not feasible. However, as the SKATES backend and modules support MPTCP, the communication between the SOCKS proxy endpoints is multi-link capable. Here, MPTCP transparently replaces plain TCP as shown in Figure 3 while the connections from the communicating processes to the proxy endpoints use regular TCP. As payloads do not need to be re-encoded, but are simply copied between TCP and MPTCP sockets, runtime overhead is minimal resulting in no significant performance degradation.

On the transport layer, MPTCP acts as a fully compatible TCP extension. The initial handshake connects to a destination address and port using the host's IP routing table. If the destination is also MPTCP capable, available secondary IP addresses are exchanged. For secondary addresses that are routable between the endpoints, additional subflows are established. This process requires that the initially provided destination address can always be connected to. Therefore, connections may fail if the primary path is unavailable, regardless of secondary path availability. To address this shortcoming, this work contributedan extension to shadowsocks that handles failed connection setups and seamlessly switches between known destination addresses. With this addition, connections can be opened when any single route is available.

Fig. 5. Photo of the evaluation equipment shows the Husky SAR robot. On top of the Husky, the communication equipment (the interoperable communication module with an LTE-Modem and Wi-Fi configuration) are attached. The 360° camera is also mounted on top, using a stand.

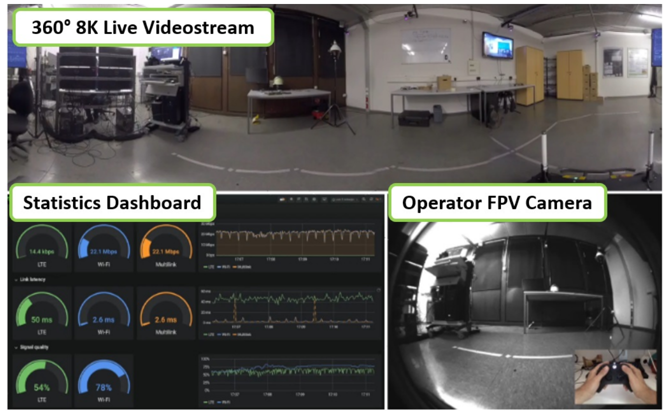

Fig. 6. Operator perspective during the experiments shows the assembled video streams and communication performance monitor. The video of the experiment can be found online at [4] Top: view of the stitched 360° live stream. Bottom left: dashboard showing the communication link's key-performance indicators (c.f. Figure 8). Bottom right: FPV camera for the teleoperated control of the robot.

C. Providing Security and Mutual Endpoint Discovery

When traversing multiple networks, especially in public LTE or public Wi-Fi, NAT can become an issue. In public deployments, mobile clients are often assigned a local address (e.g. subnet *10.0.0.0/8* for IPv4) and masqueraded to the Internet by a gateway. This means the clients are not addressable from external networks, making externally initiated TCP connections impossible. In addition, even if unique addresses are assigned (the default for IPv6), externally initiated connections may be blocked by firewalls. Furthermore, middleboxes on the Internet may remove MPTCP header extensions, effectively downgrading the protocol to regular TCP. To circumvent these issues, the system concept utilizes a VPN solution for public networks. Within the scope of this work, the VPN implementation Wireguard [16] is used, which has been merged into Linux kernel version 5.6.0. If both backend and communication modules are behind firewalls, the VPN traffic uses a server with a public IP address as an intermediate hop. This approach continues previous work of the authors in [7]. Due to the used VPN, the endpoints operated in the same local network and are able to mututally discover each other without interference. In addition, the VPN's as well as shadowsocks encryption increases general security of the robot operation and prevents external manipulation or eavesdropping.

D. Communication Link Health Monitoring

During operation, health monitoring of the communication system can help the operator avoid areas with bad connectivity or identify issues caused by degraded links. To enable detailed analysis of related metrics, the interoperable communication modules continuously gather physical layer information (e.g. signal strength) as well as traffic and latency measurements for each communication interface. This data is persisted in a central time series database, also utilizing the multi-link tunnel for the connection. Figure 4 shows a screenshot of the real-time dashboard visualization presented to the operator.

IV. Performance Evaluation

This section presents the experimental evaluation of the proposed approach and the previously described protocol extensions. The system under test is a mid-sized Unmanned Ground Vehicle (UGV) which is teleoperated through a challenging indoor environment for exploration purposes. A detailed description of the system follows, after which we conclude the section with a discussion of the experimental results.

Figure 6 shows a screenshot of the operator perspective and the finally assembled video during the experiment. A video covering the whole experiment can be found online at [4]. On the top, the 360° perspective of the robot is shown. This perspective can also be re-streamed to Virtual Reality (VR)-googles to allow rescue forced an intuitive approach to access the stream. On the bottom left real-time statistics are illustrated. Those statistics cover the throughput, latency and network quality for each RAT as well as the final multi-link of the health monitoring module. The bottom-right shows the FPV camera of the robot. This camera is mainly used by the operator to remote navigate the vehicle. In the bottom right of this perspective the operators joystick and hands are shown.

A. Search and Rescue Robot Setup and Equipment

Figure 5 shows the Clearpath Husky platform used in the experiments. The robot is equipped with various payloads for search and rescue missions. Most prominent is the top-mounted surround camera system, consisting of six wide-angle cameras. The individual camera streams are processed into one 360° video in the backend. The surround perspective enables

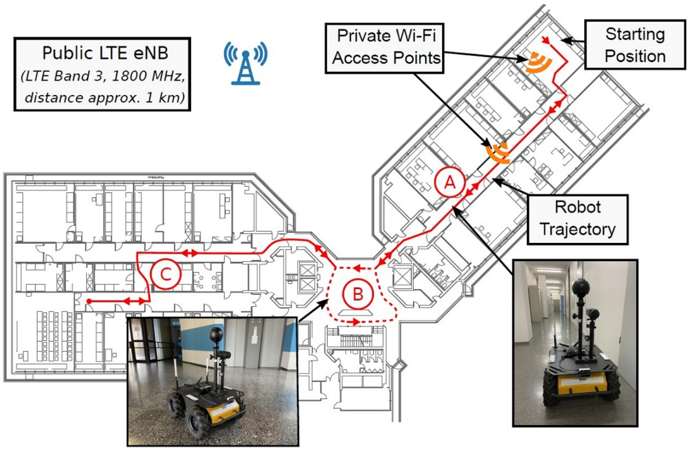

Fig. 7. Trajectory of the robot during the experiments in the challenging indoor scenario. Especially in the stairwell area Ⓑ a lot of shadowing and negative interferences occur due to the building's concrete walls. The Wi-Fi access points are located inside the building, while the public LTE cell tower is on the rooftop of a close building in an approximate distance up to 1 km. Area Ⓐ is covered by Wi-Fi, in Ⓑ signal strength is decreased, while in area Ⓒ only the LTE signal is available.

rescuers to effectively search the enviroment while remaining independent of the robot's current orientation, e.g. viewing the stream with VR headsets. Using such a camera in post disaster scenarios, enables exploration of areas which are dangerous to enter or inaccessible for humans. The camera is controlled via a proprietary Hyper Text Transfer Protocol (HTTP)-based protocol while the video streams use RTMP, illustrating the need for transparent tunneling in SKATES. In the experimental evaluation, the SKATES system makes use of one private Wi-Fi and one public LTE link. However, the concept is not limited to two RATs. Additional links such as a second public LTE operator could be added in the future. The robot is remotely controlled using ROS. For teleoperation, a dedicated FPV camera is installed at the front of the robot, interfacing with a ROS instance in the robot's on-board network. The FPV camera publishes single Motion JPEG (MJPEG) compressed video frames via ROS topics. The topics are forwarded as described in III-A. The FPV camera stream is throttled to 5 messages per second while teleoperation commands are forwarded without throttling.

B. Evaluation Scenario Description

Figure 7 shows the approximate trajectory of the vehicle inside a university building. Part of the environment, indicated by area Ⓐ, is covered by Wi-Fi. In addition, a public Mobile Network Operator (MNO) provides an LTE network with a base station in close proximity to the building. With these two communication networks available to the robot, the challenge of the scenario lies in the handover between different technologies. The stairwell area Ⓑ causes challenges for both technologies due to its massive concrete walls. The office area at Ⓒ is outside of the Wi-Fi coverage, so only the LTE network is available.

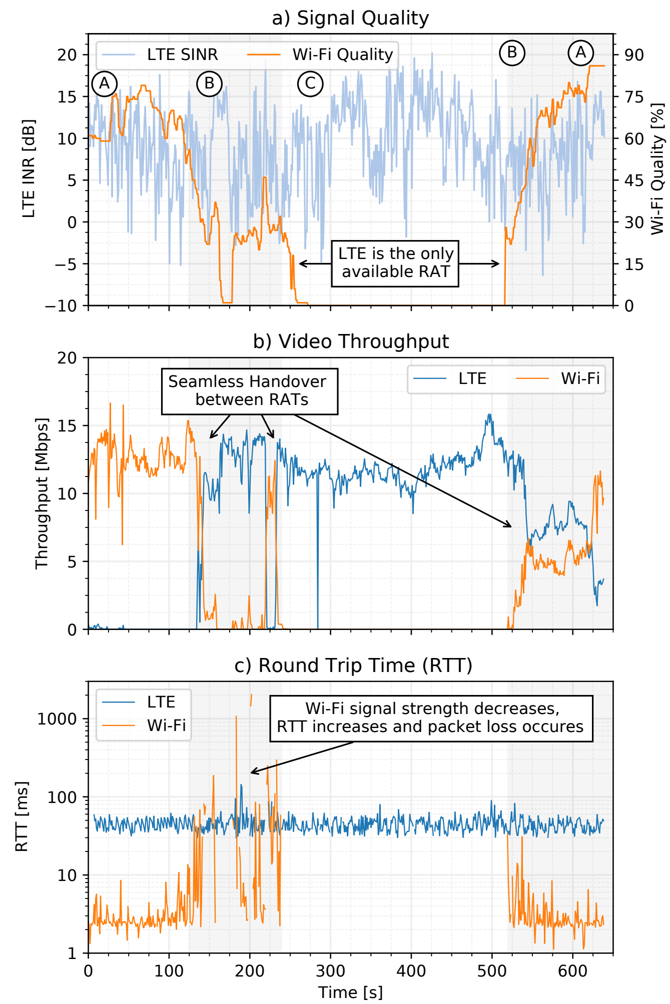

Fig. 8. Results of the experimental validation show the key performance indicators of the communication benchmarks and benefits of the Multi-RAT approach. The circled letters indicate approximate positions according to Figure 7. a) Illustrates the signal quality in form of the Signal to Interference plus Noise Ratio (SINR) for LTE and the proprietary Wi-Fi signal quality indicator. b) Throughput of the video stream for each communication link. SKATES enables smooth handovers between both RATs due to its multi-connectivity approach. c) Latency between robot and operator as RTT.

C. Experimental Results

Figure 8 a) shows an evaluation of the signal quality for each interface as the robot moves between areas Ⓐ–Ⓒ. For LTE the estimated SINR is illustrated as reported by the modem on the left side of the Figure. On the right side, the signal quality reported by the Wi-Fi module's Linux driver is illustrated as percentage. While moving from the corridor to the stairwell the Wi-Fi signal weakens (time between 125 to 175 s). In the far section of the building (see Figure 7), Wi-Fi has no coverage. The LTE signal strength is sufficient to keep a stable connection along the entire trajectory. Considering the utilized LTE modems and required throughput, SINR values above 10 dB can be considered as ideal and between 0 and 10 dB as sufficient. Below 0 dB the modem is able to stay connected to the enhanced NodeB (eNB), but throughput may be heavily degraded [7].

The video stream throughput is shown in Figure 8 b) as a time-series. The surround camera's six individual RTMP streams were configured to an approximate bitrate of 2 Mbps each, or 12 Mbps total. Actual bitrates may vary depending on the dynamics of the image data. Especially when the robot moves and video compression becomes less effective, the overall data rate of the video stream is higher. Thus, the actually measured throughput shows a high degree of variance. The resulting throughput evaluation reflects the movement of the robot and the network availability. The area of the starting position and the first corridor Ⓐ has strong Wi-Fi coverage as a line of sight to the access points can be maintained. MPTCP directs all traffic over the Wi-Fi. MPTCP uses the Lowest-RTT-First scheduler. This scheduler prioritizes the interface with the lowest average Round Trip Time (RTT). When not all data can be sent via the lowest RTT interface, data is scheduled on the interface with the second lowest RTT and so on. When the robot moves to the stairwell area Ⓑ Wi-Fi signal quality decreases. The data scheduler starts prioritizing LTE and a seamless handover is performed. In the area on the far side, Ⓒ the Wi-Fi signal is not strong enough for coverage. Therefore, the SKATES module communicates uses LTE only, while the Wi-Fi registers a full disconnect. When moving back towards the starting position, Wi-Fi signal strength increases and the communication module re-connects to the network. While moving closer to the access points, more data can be scheduled on the Wi-Fi link. As the maximum throughput of Wi-Fi is not sufficient in the transition between Ⓑ to Ⓐ, MPTCP schedules data on both RATs in a 60-40 ratio.

In the next evaluation Figure 8 c), the latency in form of RTT is investigated. Latency measurements are performed periodically using Internet Control Message Protocol (ICMP) as part of the health monitoring system. When available with good signal quality, the Wi-Fi achieves an average RTT between 1–5 ms. Latency in LTE falls between 30 and 60 ms, which includes a VPN-hop via the public Internet as described in III-C. In the transition area, Wi-Fi RTT increases significantly due to packet retransmissions made necessary with degrading signal quality. With the signal weakening further, packet loss occurs which is visualized in the Figure by gaps in the RTT lines.

Finally, Figure 9 presents a statistical evaluation of the throughput in form of a Cumulative Distribution Function (CDF). LTE contributed a higher throughput and was used approx. 82 % of the time, while Wi-Fi has been used only 49 %. On average (median value) LTE was at 11 Mbps, while the total multi-connectivity approach matched the video throughput of 12 Mbps. The video could be reliably transported throughput the whole experiment.

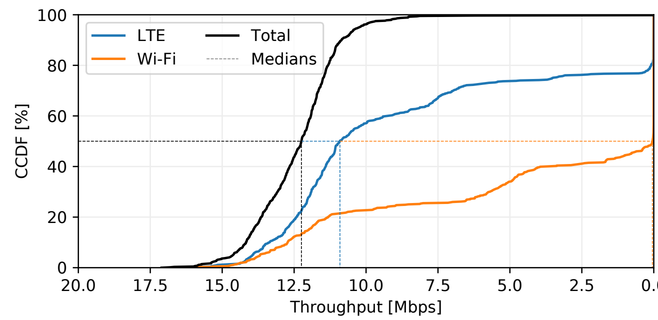

Fig. 9. The Cumulative Distribution Function (CDF) of the throughput shows the percentage of how the throughput reached. Both LTE and Wi-Fi were continuously used (neither achieved 100%). LTE contributed a higher throughput. Generally, the RTT below 100 ms is ideal for real-time robot control.

V. Conclusion

The underlying work presents and evaluates a Multi-RAT approach for maximizing communication reliability for SAR robots in challenging environments. The concept was implemented and evaluated experimentally in a test mission. The testbed comprised a robot equipped with an 8K surround camera as its main payload and an interoperable communications module with Wi-Fi and LTE modems. In the experiment, the robot was teleoperated through a building and successfully streamed high quality image data of the environment. The video stream as well as the control data was reliably transmitted throughout the duration of the experiment. Inter RAT handovers were handled seamlessly and with no negative impact on performance. In future, the authors plan to equip the robot with 5G and Wi-Fi 6 communication hardware and perform larger scale field tests in environments specifically designed for rescue robotics communication benchmarking.

Acknowledgment

This work has been supported by German Federal Ministry of Education and Research (BMBF) for the project A–DRZ (Establishment of the German Rescue Robotics Center, 13N14857).

References

[1] J. Delmerico, S. Mintchev, A. Giusti, B. Gromov, K. Melo, T. Horvat, C. Cadena, M. Hutter, A. Ijspeert, D. Floreano, L. M. Gambardella, R. Siegwart, and D. Scaramuzza, "The current state and future outlook of rescue robotics," *Journal of Field Robotics*, vol. 36, no. 7, pp. 1171–1191.

[2] B. Hesmans, F. Duchene, C. Paasch, G. Detal, and O. Bonaventure, "Are TCP Extensions Middlebox-Proof?" in *Proceedings of the 2013 Workshop on Hot Topics in Middleboxes and Network Function Virtualization*, ser. HotMiddlebox '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 37–42. [Online]. Available: https://doi.org/10.1145/2535828.2535830

[3] Clowwindy, Madeye, and Linusyang, "shadowsocks-libev version 3.3.4," https://github.com/shadowsocks/shadowsocks-libev/tree/v3.3.4, jan 2020, accessed on 2020-06-24.

[4] Communication Network Institute (CNI), "Video recording of the experiments documented in this paper," https://www.youtube.com/watch?v=hHqXNUFUUG4, jun 2020.

[5] A. Rao, M. Visali, S. Shailendra, B. Panigrahi, and A. Simha, "Reliable robotic communication using multi-path TCP," in *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*, 2017, pp. 429–430.

[6] J. Güldenring, L. Koring, P. Gorczak, and C. Wietfeld, "Heterogeneous Multilink Aggregation for Reliable UAV Communication in Maritime Search and Rescue Missions," in *15th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2019) - Sixth International Workshop on ICT Systems for Public Protection and Risk Reduction - 2019 (ICT4PPRR'19)*. Barcelona, Spain: IEEE, Oct. 2019.

[7] J. Güldenring, P. Gorczak, F. Eckermann, M. Patchou, J. Tiemann, F. Kurtz, and C. Wietfeld, "Reliable Long-Range Multi-Link Communication for Unmanned Search and Rescue Aircraft Systems in Beyond Visual Line of Sight Operation," *Drones*, vol. 4, no. 2, 2020. [Online]. Available: https://www.mdpi.com/2504-446X/4/2/16

[8] T. Viernickel, A. Froemmgen, A. Rizk, B. Koldehofe, and R. Steinmetz, "Multipath QUIC: A Deployable Multipath Transport Protocol," in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–7.

[9] B. Han, F. Qian, L. Ji, and V. Gopalakrishnan, "MP-DASH: Adaptive Video Streaming Over Preference-Aware Multipath." New York, NY, USA: Association for Computing Machinery, 2016.

[10] S. Song, J. Jung, M. Choi, C. Lee, J. Sun, and J. Chung, "Multipath based adaptive concurrent transfer for real-time video streaming over 5g multi-rat systems," *IEEE Access*, vol. 7, pp. 146 470–146 479, 2019.

[11] S. Shailendra, K. Aniruddh, B. Panigrahi, and A. Simha, "Multipath tcp path scheduler for drones: A segregation of control and user data." New York, NY, USA: Association for Computing Machinery, 2017.

[12] "Pound: A multi-master ROS node for reducing delay and jitter in wireless multi-robot networks," *Robotics and Autonomous Systems*, vol. 111, pp. 73 – 87, 2019.

[13] S. K. Saha, A. Kannan, G. Lee, N. Ravichandran, P. K. Medhe, N. Merchant, and D. Koutsonikolas, "Multipath TCP in Smartphones: Impact on Performance, Energy, and CPU Utilization." New York, NY, USA: Association for Computing Machinery, 2017.

[14] Q. De Coninck, M. Baerts, B. Hesmans, and O. Bonaventure, "A First Analysis of Multipath TCP on Smartphones," in *Passive and Active Measurement*, T. Karagiannis and X. Dimitropoulos, Eds. Cham: Springer International Publishing, 2016, pp. 57–69.

[15] M. Schwarz, T. Rodehutskors, D. Droeschel, M. Beul, M. Schreiber, N. Araslanov, I. Ivanov, C. Lenz, J. Razlaw, S. Schüller, D. Schwarz, A. Topalidou-Kyniazopoulou, and S. Behnke, "NimbRo Rescue: Solving Disaster-response Tasks with the Mobile Manipulation Robot Momaro," *Journal of Field Robotics*, vol. 34, no. 2, pp. 400–425.

[16] J. A. Donenfeld, "WireGuard: Next Generation Kernel Network Tunnel," in *Network and Distributed System Security Symposium (NDSS)*, 2017.