Gerwien, Manuel; Großmann, Marcel; Krieger, Udo R.

The System Architecture of a Reliable Telesurgery Service and its Performance Analysis

Date of secondary publication: 27.04.2026

Accepted Manuscript (Postprint), Conferenceobject

Persistent identifier: urn:nbn:de:bvb:473-irb-114838x

Primary publication

Gerwien, Manuel; Großmann, Marcel; Krieger, Udo R. (2024): The System Architecture of a Reliable Telesurgery Service and its Performance Analysis, in: F. Phillipson, G. Eichler, C. Erfurth, G. Fahrnberger (Ed.), Innovations for Community Services : 24th International Conference, I4CS 2024, Maastricht, The Netherlands, June 12–14, 2024, Proceedings, Cham: Springer Nature Switzerland, pp. 257–274, doi: 10.1007/978-3-031-60433-1_15.

The System Architecture of a Reliable Telesurgery Service and its Performance Analysis

Manuel Gerwien, Marcel Großmann, and Udo R. Krieger(✉)

Fakultät WIAI, Otto-Friedrich-Universität, An der Weberei 5, 96047 Bamberg, Germany

udo.krieger@ieee.org

Abstract. Today, new smart e-health applications are enabled by the very rapid evolution of high-speed networking and cloud computing technologies. This development of new communication and computing techniques as well as the incorporation of machine learning for intelligent image analysis offer new opportunities for classical medical services like telesurgery. In our paper we consider the design of a classical telesurgery system and analyze its architecture. Then we discuss the extension of its functional modules to support remote surgery based on a reliable multi-path communication among its components. We provide an investigation of important key performance indices of such a remote-controlled telesurgery service. To achieve deeper technical insights on the system's performance of such a distributed architecture and its communication flows, a telesurgery prototype is studied by GNS3-emulations within a virtualized test bed. In this way we investigate the fundamental computing and network performance indices of our prototype.

Keywords: e-Health · Telepresence · Telesurgery · Performance analysis

1 Introduction

Today, new advanced Internet-of-Things services like smart e-health applications are enabled by the very rapid evolution of high-speed networking technologies which incorporate 5G and 6G wireless networks, software-defined networking and network function virtualization as building blocks, cf. [8,9,16]. The latter communication infrastructure constitutes the basis for cloud computing, fog computing, and multi-access edge computing and offers the opportunity to integrate blockchain technology for medical applications, cf. [1,3,14]. The fast development of these advanced communication and cloud computing techniques and the incorporation of new machine learning methodologies for intelligent image analysis also offer new opportunities to improve classical medical services such as telesurgery and neurosurgery, cf. [12,15,16].

In our paper we consider first the technical architecture of a classical telesurgery system. Then we discuss the extension of its components to support remote-controlled surgery based on advanced communication networks. We are interested in investigating the impact of distortions within such a network onto the system performance and study important key performance indices (KPIs) of the telesurgery system. To achieve such deep technical insights, the architecture and communication infrastructure of a prototypical telesurgery system is emulated by a virtualized test bed using a GNS3 network simulator. By these means we investigate the computing and network performance indices of our prototypical telesurgery architecture and identify its weaknesses.

The paper is organized as follows. In Sect. 2 we discuss the system architecture and networking functionality of a distributed, remote-controlled telesurgery infrastructure. Then we describe the emulation of a virtualized prototype using the GNS3 network simulator in Sect. 3. We also present experimental results on the penetration of major performance metrics due to distortions in the underlying communication network of our telesurgery system. Finally, some conclusions are presented in Sect. 4.

2 System Architecture of a Remote-Controlled Telesurgery Infrastructure

In this section we analyze the HW/SW-architecture of a classical telesurgery system and identify its basic building blocks. The study is inspired by existing telepresence systems and remote-controlled neurosurgery infrastructures. In particular, we look at the architecture of modern angiography systems like Artis, that are employed for tele-imaging during neurosurgery processes in advanced e-health scenarios, cf. [18]. Our study also reflects on related studies of remote-controlled telesurgery, cf. [2,15,17,19,20].

From a general perspective, a remote-controlled telesurgery system can be divided into three basic building blocks shown in Fig. 1. At the *patient side* on the lhs of Fig. 1, the system comprises *local components* for the medical imaging and treatment of a patient. The patient lies in the center on a patient table. The local staff can move the table by a local control panel. The system includes medical robot functionalities and X-ray arms for tele-imaging, e.g., to provide fluoroscopy images. The X-ray arms can be moved and radiation can be triggered by the same control panel. As soon as an X-ray beam has been triggered, the creation of a fluoroscopy image is initiated. A detector unit captures the X-rays and

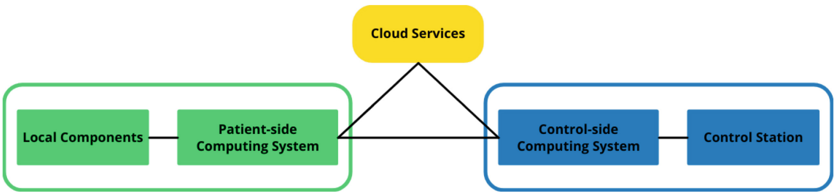

Fig. 1. Basic components of a remote-controlled telesurgery system

imaging algorithms reconstruct the patient's vessel structures. Then the medical images can be viewed on a local screen. If images are frequently triggered in a short amount of time, a related video can be generated and used, for instance, to analyze the function of a heart or other organs. In addition a telepresence system as a user interface for the local medical staff can be incorporated. The *patient-side computing system* also provides the logical endpoint to compute the required control actions for the actuators of the telesurgery units such as the robot arms for a coronary artery treatment, for instance, or the X-ray arms of medical imaging units. These components can execute commands which are triggered by control messages sent from the *remote-control side* on the rhs of Fig. 1.

At this *remote-control side* a *control station* and the *control-side computing system* are the basic functional blocks of the system architecture. They provide the user interfaces for tele-imaging, telepresence, the patient's table movement, and the robotic actuators' control in a joystick like fashion.

In a classical medical treatment the medical operation takes place locally and the physician is near the patient. In the remote-controlled scenario an additional physician resides in a separate location outside the patient's site and controls the telesurgery system by her remote actions after an imaging analysis. In such a scenario the incorporated medical images such as fluoroscopy images depict the patient's status on a remote display.

The *cloud services* are employed to manage the connectivity and the synchronization between both parts of the telesurgery system and its related functional blocks during medical actions. When a logical and physical connections between both sides of the distributed system have been established, the system status, logging messages, and network management information are exchanged. For this purpose a communication system with a TCP/UDP/IP protocol stack including security functionality based on TLS/DTLS and VPN protocols is employed.

The resulting technical structure and the information flows of such a remote-controlled telesurgery infrastructure are illustrated in Fig. 2.

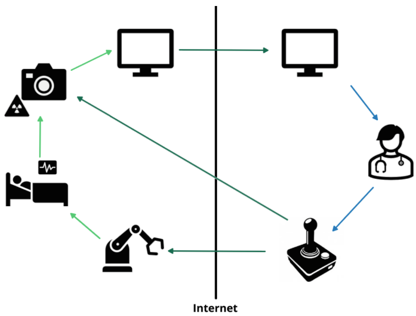

Fig. 2. Technical structure of a remote-controlled telesurgery system and the information flows between its functional components

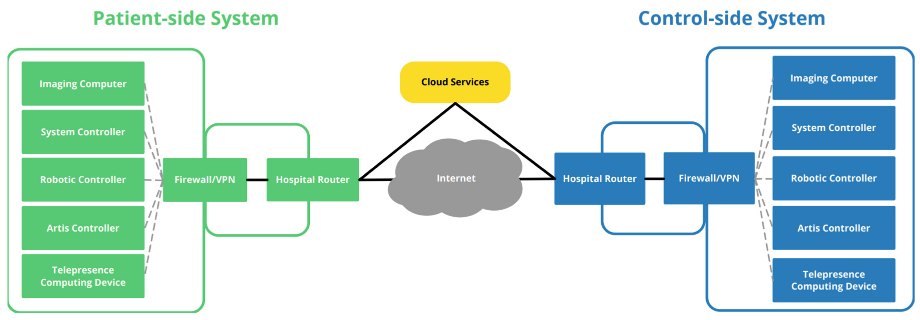

Fig. 3. Basic functional blocks of a prototypical telesurgery system

2.1 Basic Functional Modules of a Prototypical Telesurgery System and Its Information Flows

Now we identify the basic functional modules and information flows within a prototypical telesurgery system shown in Fig. 1. Regarding the patient side, the incorporated technical components of the telepresence and neurosurgery units work independently when a physician is locally available as decision maker at the patient's table. If the remote control via a network connection is instantiated, then all functional components available locally must also be offered remotely. The related data must be transferred between the patient side and the control side, too.

The basic functional modules of a prototypical telesurgery architecture and the required communication units between them is illustrated in Fig. 3, cf. [18]. Considering the green boxes at the patient side on lhs of Fig. 3, the associated local components consist of all the equipment that is necessary for a local treatment. It includes the robotic units, the medical imaging system, a telepresence system and a user interface (UI) for the local staff to control the system. In our prototype we focus on the Artis-system [18] with fluoroscopy images as tele-imaging system. Then this patient's table unit together with the associated computing system can be divided into six subcategories:

- The *imaging computer* preprocesses the captured medical images of the patient, e.g., fluoroscopy images during an angiography operation. In the latter case desired tissue features are highlighted on the local screen and the physician can navigate a guidewire cleanly within the vessel walls of the patient. The latter image data are encoded, compressed and transferred to the remote side by IP-packet streams.
- The *system controller* is the central computational unit. It provides the link between the neurosurgery system and the medical expert via the UI. It also manages the system configuration, the fault handling, the logging of system states and actions as well as the local communication of the system and the connections both to the remote side and to the cloud. To communicate with the remote side, the system controller obtains address information and configuration settings from the cloud services. All data transmitted to a remote side runs along this unit at first.

- The *robotic controller* represents the robot unit of the system. It executes the commands received from the remote side. It consist of a robotic actuator component mounted on the patient's table and a controller for processing incoming control signals and the actuation of the robotic motors as well as for the required local calculations of steering tasks.
  The system and robotic controllers form a unified technical unit in a locally controlled system. Regarding a remote operation, they are expanded by a communication management system and two separated communicating blocks to create a patient side and a remote side system.
- In our setup the employed X-ray arms can be navigated by an *Artis controller*. Depending on the used neurosurgery system, there can be two arms, one in horizontal direction and one in vertical direction. A physician can move these arms to his favorite position and start with the fluoroscopy imaging of a patient. In an angiography scenario, for instance, these resulting images are very important for the physician because she safely navigates a guidewire along the patient's vessel walls based on these medical images.
- The staff's communication between the patient and control side can be established by means of *telepresence computing devices* at both sides. They provides a bi-directional connection. At the patient side one or more room cameras are used normally to have different views on the patient-side system. It must be possible for the medical doctor to see which person is located at what place in the treatment room and where the patient-side operators are. This view is very important for the steering of X-ray beams and the movement of the X-ray arms since X-rays are harmful to humans. Hence, verbal communication must be supported in addition to the video transmission. Then physicians at both sides can exchange information about the current status and the planning of further treatment steps.
- The telesurgery system is supported by a *communication system* which carries IP-encapsulated data traffic along the Internet. It has two building blocks, namely the *hospital routers* and the *hospital network infrastructure*.

The *cloud services* provides three major functions to manage the entire family of devices, their users and the connections between the patient and control side:

- First, a *service for device and user management* is used to manage all users of the system and their privileges, to keep track of all active devices, to configure the robotic networks, to support remote servicing, and to perform software updates. When a new patient-side system is going online, the device management service has to register the status of these devices and their availability for the control-side systems.
- The second task is given by a *session management service*. It is responsible for establishing and managing a control-signal connection between the patient side and the remote side.
- The third task concerns a *network management service* which monitors all systems and their status. When an active connection between a patient side and a control side is established, it controls the availability of both systems, monitors the Quality-of-Service (QoS) of the connections, and collects related

usage and performance data. It should also be possible to perform diagnostic network and system tests.

2.2 Communication Infrastructure of a Reliable Telesurgery Service

In this section we briefly describe in more the network architecture and communication structures of our reliable telesurgery system. The communication architecture comprising the hospital routers and the hospital network infrastructure toward the cloud services has been depicted in Fig. 3. In our work we will not focus on the hospital routers and their IP-network in detail since the network infrastructure is set up differently in each hospital and managed by different service providers. Therefore, it is not possible to make a general statement about the error behavior and security issues due to DDoS penetration. Here we assume that the related firewall/VPN unit is directly connected with the Internet. Therefore, we distinguish the following components:

- A *firewall/Virtual Private Network (VPN) controller* constitutes the security interface of the patient side and control side toward the hospital routers. It shall protect the system from cyberattacks and other malicious packet traffic and can establish dedicated connections by a VPN controller. It also creates authorized, encrypted end-to-end IP-connections between the patient side and the remote side or to a cloud services system for management and maintenance purposes.
- The *hospital routers* establish the IP-paths within the distributed telesurgery system and toward the cloud computing infrastructure. They also provide standardized network and traffic management functionalities for IPv4- and IPv6-packet flows.

From a network perspective the entry point into each part of the telesurgery system at the patient side and control side is provided by a router with firewall functionality. To guarantee a high availability of the connections between the patient- and control-side systems, a multi-path concept should be supported among these sides. Therefore, we have employed Fortinet's FortiGate-60E routers with firewall and SD-WAN functionalities in our prototype, cf. [4]. The latter communication system also supports the establishment of a primary and a secondary path among the routers within an SD-WAN zone shown in Fig. 4.

Here the physical network interfaces are represented as black boxes within the two different data paths shown as pipelines. These connections are defined as SD-WAN members. The task of an SD-WAN zone is to evaluate availability checks at each member of the zone. During the evaluation phase the latency, packet loss and jitter of each member are measured. The result of those checks are used in appropriate SD-WAN rules. In this way, the system can monitor and manage the status of the instantiated physical transport connections. Then it can select the best path for the data transmission to satisfy the required QoS/QoE specifications. To ensure a secure transport of the information between both FortiGate routers a VPN tunnel is created along each network path and the data are encrypted by an IPsec tunnel mode, cf. [7].

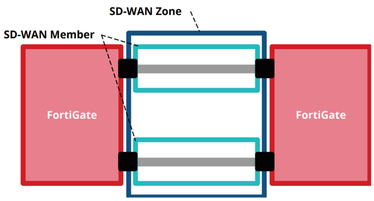

Fig. 4. Interconnection of two FortiGate routers within a single SD-WAN zone

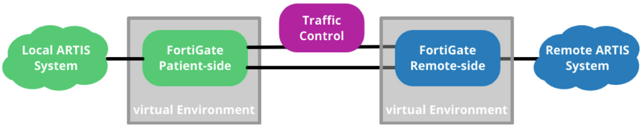

Fig. 5. Communication architecture of two virtualized FortiGate routers embedded into the local and remote Artis system

In our developed prototype these FortiGate routers are integrated into the Artis system as virtualized entities by means of virtual machines, see Fig. 5. To support the manipulation of transferred IP-packet streams during our experiments, e.g., by delaying or dropping packets, a traffic control block has been realized along the first IP-path employing the Linux traffic control function *tc*, see [11].

As the telesurgery system acts in a closed loop mode, the required information flows must be realized in real-time. At the patient side of an angiography action, for instance, the Artis system acquires fluoroscopy images from the patient's body in a specified frequency between one to sixty frames per second. In the remote-controlled setup the release of a X-ray beam to acquire new fluoroscopic images is started by the physician at the remote-control side. The captured images are first computed locally and the resulting video stream of successive fluoroscopy images is shown as a stream on the local display. Then the video signal of the display is captured, encoded as H.265 data set of tele-images and transmitted to the control-side system. There it is decompressed and displayed. Then the remotely acting physician can see the live X-ray images at her control-side display. Based on these images she can control the movement of the robot arms at the patient side with a special console. When she steers one joystick into a direction the associated signal of this movement command is transmitted to the patient-side system. Then the local robot can perform the specified action, e.g., move a used guidewire forward within the vessel walls of the operated patients or rotate the guidewire. Then the movement of the guidewire is captured again by new fluoroscopy images. The latter are transmitted back to the physician at the remote-control side. Such a prototypical workflow with its related information flows between the basic building blocks of the telesurgery system is depicted in Figs. 2 and 6.

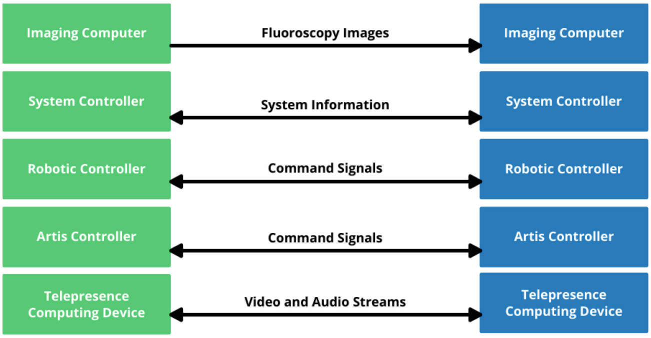

Fig. 6. Information flows between the basic components of the telesurgery system

Using the classical Diffserv QoS-model of IP-networks, cf. [9], the system's information flows and the command-signal flows can then be mapped to the real-time priority QoS-traffic class Expedited Forwarding whereas the tele-imaging and telepresence data streams may be assigned to the high priority QoS-traffic class Assured Forwarding. In our current study we have not been focusing on these assignment issues of optimal QoS-traffic classes.

2.3 Performance Requirements of the Communication Architecture

Regarding the performance requirements of a prototypical telesurgery system we first consider the required bandwidth of the tele-imaging and telepresence system along the logical transport channels between the patient and the remote-control side on the uplink and on the opposite downlink path. Inspired by the Artis system [19] we suppose that three different cameras at the patient-side with 3 Mbps per camera generate a capacity demand of the UDP/IP-stack of 9 Mbps along the uplink. Assuming one camera at the remote-side generates a demand of 3 Mbps along the downlink.

Based on the analysis of typical tele-imaging system we assume that the demanded uplink capacity demand is given about 40 Mbps and the downlink capacity demand is about 4 Mbps. Practical studies reveal that robotic- and system-control traffic demands a symmetric capacity of about 5 kbps, whereas a typical Artis controller needs 200 kbps. In summary, the complete bandwidth requirements of all logical channels of our five components shown in Fig. 6 are summarized in Table 1.

Considering the packet delay from the patient side towards the remote-control side and its associated jitter between the transferred IP-packets, one can assume that jitter values below 700 ms are not vulnerable to a proper operation of the distributed tele-imaging, telepresence and robotic-control components of the systems. Regarding the end-to-end (E2E) delay T_l between the critical tele-imaging or telepresence modules in the distributed telesurgery system, one has to take into account the transmission delay and signal propagation delay along

Table 1. Typical bandwidth requirements of each functional component

Functional Component	Uplink	Downlink
Imaging Computer	40 Mbps	4 Mbps
System Controller	5 kbps	5 kbps
Robotic Controller	5 kbps	5 kbps
Artis Controller	200 kbps	200 kbps
Telepresence Device	9 Mbps	3 Mbps

an IP-path between the routers and their firewalls T_t, and the processing and computing latency in the related functional blocks on top of the IP-layer at the patient and remote-control side T_{ap} and T_{ar}, in total $T_a = T_{ap} + T_{ar}$. If an additional random delay T_i is occurring along the IP-path and if other random effects are ignored, one gets the following E2E-latency formula:

$$T_l = T_a + T_N = T_a + T_t + T_i = T_{ap} + T_{ar} + T_t + T_i$$

Previous studies [10,12,13] have shown that the network induced latency component $T_N = T_t + T_i$ should not exceed 300 ms to guarantee a proper remote-controlled operation of a telesurgery system.

Considering the tele-imaging system, we expect that it encodes medical images by the H.265 standard. Then we can assume that 2% IP-packet loss can be tolerated, see [6,21]. If this value is guaranteed by the communication system along the IP-connections, the required QoS-guarantee of telepresence is also not affected.

3 Emulation of a Virtualized Prototype Offering a Reliable Telesurgery Service

In this section we discuss the emulation of a virtualized prototype offering a reliable remote-controlled telesurgery service based on the system architecture sketched in Sect. 2. The SD-WAN network architecture shown in Subsect. 2.2 is emulated by means of a virtualized test bed applying the tool GNS3, see [5].

3.1 Network Architecture of the GNS3-Prototype

The system architecture of our investigated telesurgery prototype is depicted in Fig. 7. The emulated network architecture of the distributed telesurgery system is directly integrated into the employed Artis-controllers using virtual machines. Using a router virtualization, the related network structure between both sides is depicted in Fig. 8.

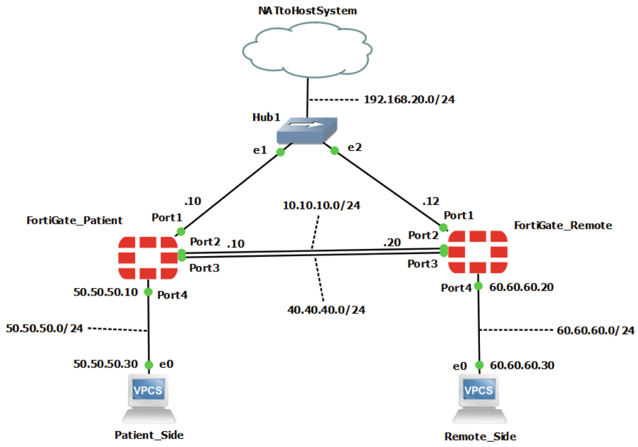

Fig. 7. GNS3-prototype of an emulated telesurgery service with two virtualized Forti-Gate routers

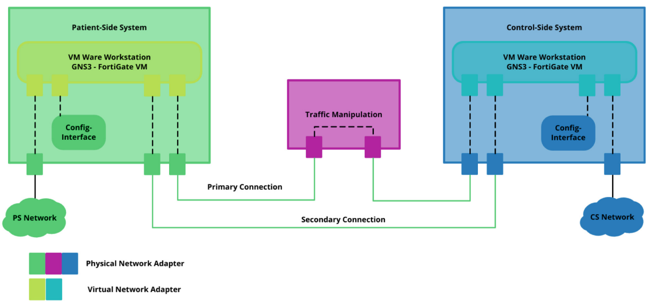

Fig. 8. Integration of virtualized FortiGate routers into the Artis-controller system

The study of our prototype has three objectives. First, the functionality of the Artis-controller which gets access to the session and connection management functionality to steer the SD-WAN IP-network will be tested. Secondly, QoS-requirements on the network connectivity shall be determined that guarantee a proper and reliable operation of the remote-side functionality of the Artis-controller. It has to guarantee the robustness of the system against network distortions. In particular, we want to analyze the performance impact if an additional latency component, a packet loss or a reduced bandwidth is imposed along the used data transport paths. In this manner it is possible to use the prototype for other user studies and one can derive threshold values for KPIs of the system more precisely and gain additional QoE-feedback by potential users. It is the third objective to evaluate the SD-WAN functionality of the FortiGate routers and to analyze how the path redundancy affects the functionality of the system when a switching from the primary to the secondary connection occurs.

In a further experimental step the virtualized FortiGate routers may be replaced by real hardware routers.

For the manipulation of the transmitted data traffic and to test a decreased network quality in the Artis-controller at the remote side a computer running LINUX is introduced in the primary path between both FortiGate routers. On this system two physical network interfaces are bridged to forward network traffic. This traffic stream can be manipulated by the Linux traffic-control functionality of *tc* [11].

Our virtualized test bed comprises two virtual machines which emulate the medical tele-imaging system and the Artis-control system at the patient side and the remote-control system at the opposite remote side as well as an additional virtualized host as cloud computing side. Both building blocks at the patient and remote-control side are interconnected by two virtualized SD-WAN routers which incorporate traditional TCP/IP-based connectivity and VPN functionalities of commerical high-speed routers for secure cloud communication such as the FortiNet SD-WAN multi-WAN-router FortiGate-60E, cf. [4].

In our prototype the latter routers establish a secure multi-path connection between the virtualized computer systems at the remote-control side and the patient side. To simplify the communication test bed, a primary connection and only one secondary connection as backbone path have been established. Furthermore, they are used to interconnect both computing systems to a centralized cloud. The latter block provides the basic remote control and orchestration functionalities within the emulated telesurgery system. In our virtualized test bed each deployed FortiGate-60E incorporates a web server. It is used to handle the required configuration tasks of the system architecture.

The cloud computing system on the top of the realized communication network in Fig. 7 provides a global API interface to the network interfaces of the SD-WAN routers. Its configuration tasks are handled by a virtual machine management system applying VMware [22].

In our prototype the cloud infrastructure of the organisation offering the telesurgery service is bound to a hub router interconnecting it to both SD-WAN routers within a local address space given by *192.168.20.0/24*. The latter is configured by a NAT functionality. Both FortiGate routers are connected to this hub and the associated local NAT network. In this way both FortiGate router are inside the NAT network to host the cloud system as management system for the configuration of both FortiGate routers and their integrated web servers. They are able to send Internet-Control-Message-Protocol (ICMP) ping messages to a specified destination for availability testing.

The network *50.50.50.0/24* represents the network on the patient-side system. *60.60.60.0/24* represents the network for the remote-side system. Both virtual machines are configured according to their associated network addresses. This means, for example, that the *Patient_Sides* network interface has the configured IP address *50.50.50.30*. Both FortiGate routers have four configured network interfaces. The first one on *Port1* represents the interface to the NAT network to reach the host system. *Port2* is the primary connection between both

FortiGate routers with *10.10.10.0/24* as network address and *Port3* is the secondary connection with *40.40.40.0/24* as network address. The last one, *Port4*, is the network interface for the hospital network. This prototype architecture is able to show that both virtual computers can reach each other with ICMP messages. Both FortiGate routers can supervise the primary and the secondary connection between each other. They can also determine if they prefer to route the whole traffic over the primary IP-path. When the primary connection reveals a bad QoS-quality or the link is completely down then the traffic shall be routed via the secondary network connection based on a trigger by the VPN-controller.

3.2 The Impact of Operational Distortions in the Communication Network on the System's Performance

By our emulation of a prototypical telesurgery architecture we want to investigate the impact of operational distortions in its communication network on the system's performance. For this purpose we focus on major key performance issues that are related to the network specification and the related service-level objectives of the telesurgery service.

The executed experiments are concerned with the performance impact of following effects on three a priori specified key performance indices (KPIs):

- Limiting the available bandwidth of logical channels transferring the user data and signaling traffic
  A logical channel for the user traffic carries video streams uplink between the local tele-imaging system and the remote-control system. Along the downlink path in the reverse direction signaling traffic is transported. We investigate the effect of bounding the required bandwidths of those channels to less than 100% of their available capacity. In our measurement study the reduction of the granted channel bandwidth will be downgraded in steps of 10 kbps. As the data rates of the telepresence streams are lower in both directions, we will focus on the tele-imaging component.
- Changing the latency along the transport channels between the SD-WAN routers
  Following the line of reasing in [13], we investigate the impact of an increasing latency along the transport channels on the end-to-end delay and the performance of the emulated telesurgery system. For this purpose we incorporate additional transport delays in the router network in the range $[0, 1000]$ ms in discrete steps of 0, 150, 250, 400, 600 and 1000 ms. It is the objective of these tests to find out the maximal tolerable end-to-end delay such that the Artis remote-control system is still operating in a proper way and the additional latency is not perceived at the video-decoder side. In our future research those tests may be repeated to monitor the quality-of-experience realized by physicians.
- Evaluation of the SD-WAN multi-path functionality
  The SD-WAN functionality of the FortiGate routers offers a countermeasure to cope with a failure of an instantiated IP-path and the related routers

between the medical imaging source and the remote-control unit. Offering the possibility to integrate a redundant second IP-path triggered by measurements, the availability of the telesurgery service can be increased and the identified path with the better KPIs may be used for the data transport. In our experiments we evaluate how this functionality behaves when the connection is switched over from the primary to the secondary IP-path. We are interested whether this switching process can be noticed by a working physicians and how long it takes until a decreased quality level of an IP-connection is detected.

These performance issue of the SD-WAN functionality will be evaluated on the emulated prototypes by our following measurement studies.

3.3 Performance Results of the GNS3-Experiments

We have performed three different series of experiments to investigate the distortion effects when the data traffic is delivered along the first IP-path based on a connection in our test bed between the functional modules of the Artis-controller at the patient side and the remote-control side.

In the first setting of the GNS3-experiments we study the performance impact when the channel bandwidth of the Artis-command signaling from the remote-controller toward the patient side is limited at the IP-layer. An encryption by IPSec has been disregarded in these experiments. It can be seen that limiting the throughput at the intermediate tc-controller to 180 kbps can guarantee a proper teleservice operation while a limit of 160 kbps induces at freezing effect at the controlled Artis-actuators at the patient side.

The experiments illustrate that a traffic priority concept using a Diffserv QoS-model along the IP-path must be realized to guarantee a proper interaction of the distributed telesurgery service infrastructure.

When the bandwidth of the logical channel of the Artis-controller from the patient side to the remote side is reduced between the related Fortigate routers, the behavior of the telesurgery service is different. The bandwidth can be reduced to 50 kbps without any effect on the controllability of the actuator. At 40 and 30 kbps an activity of the Artis components is strongly disturbed and at 20 kbps the proper operation of the system has stopped.

In the second setting of the GNS3-experiments we look at the end-to-end delay between the patient side and the remote side components and monitor the induced round-trip delay in the test bed. For this purpose the time between a deflection of the joystick at the remote side and a detection of this movement within collected fluoroscopy images at the remote side is determined. In addition, it is estimated how long it takes until the command of the joystick set the Artis actuators in motion. For this purpose an external camera has been used which filmed the control panel, the patient's Artis actuator components and the transmitted video on the display at a rate of 240 frames per second. The transmission delay between the display of the local fluoroscopy image and the display on the remote side was 50 ms. In the setup 10 test runs have been

Classes:	Frequency
500	0
600	2
700	4
800	2
900	1
1000	1
1100	0

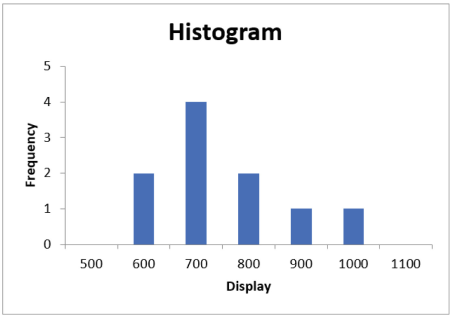

Fig. 9. Evaluated latency in ms between deflecting the joystick until the resulting movement effect is seen on the remote-side display

Classes:	Frequency
400	0
500	3
600	5
700	0
800	1
900	1
1000	0

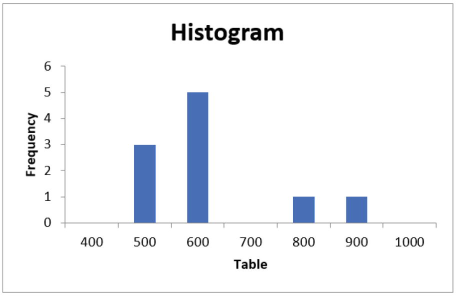

Fig. 10. Evaluated latency in ms between deflecting the joystick until the patient's table starts moving

performed. The resulting average time until an object motion has been seen on the fluoroscopy image is 705 ms with a standard deviation of 106 ms. The standard deviation is that high because a fluoroscopy image has been generated only every 100 ms, i.e., at 10 frames per second. Therefore, a change in the fluoroscopy image on the recorded video can only be determined with an accuracy of 100 ms. The evaluation of the latency measurements is depicted by a histogram in Fig. 9.

It has been harder to evaluate the duration until the Artis-actuator with the related patient's table starts moving because it is moving slowly and its very small movement cannot be detected properly by our video recording. Related experiments with 10 test runs have generated an average delay of 561 ms regarding that action. The standard deviation is given by 114 ms. These measurement results can be seen by the histogram in Fig. 10.

In Fig. 11 we illustrate the monitored latency profile of an Artis-controlled teleservice interaction between the joystick at the remote-control side, the start of a movement of the patient's table, and the aggregated latency along the whole control path. The duration of an image transfer from the local display at the patient's side to the display at the remote-control side is given by 50 ms

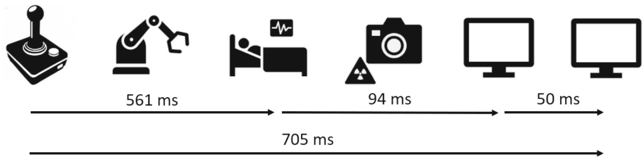

Fig. 11. Monitored latency profile of an Artis-controlled teleservice interaction

if an additional transport delay among the related routers is disregarded. The delay for a triggerd acquisition of a new fluoroscopy image at the patient's side is calculated as 94 ms. It yields an E2E-delay starting with the trigger signal of the remote joystick until the recorded image shows the new status of the moved Artis actuators on the remote display within a round-trip time of 705 ms. Disregarding the distance-based transport delays of the control messages within the underlying IP-network, an average activation of $T_a = 700$ ms can be used as reference value of the studied telesurgery service from a start of the action by a physician, e.g., by deflecting the joystick, until the verification of this activity on the remote display.

In the third setting of the GNS3-experiments we have studied the impact of an induced transport delay along the IP-connection between the FortiGate routers at the patient side and the remote side. Both directions are considered separately in these latency tests. Regarding the downlink path from the remote side toward the patient side an additionally transport delay of 150 ms could not disturb the controllability of the Artis subsystem. A latency value of 250 ms could already be properly perceived, but it did not destroy the controllability requirements of the telesurgery system. At a delay value of 400 ms the patient's control-table setting was noticeably affected, whereas the table movement became uncontrollable for latency values of 600 ms, when the joystick is deflected sharply.

Regarding the uplink connection from the patient side toward the control side, an operational deterioration of the system could not be perceived if a transport delay in the range $[0, 1000]$ ms has been introduced. The reason is that in our current Artis-prototype only the TCP-acknowledgment messages arising from the control signals of the Artis-system are transmitted uplink via the FortiGate routers. This message transfer is not so much penetrated by the induced delay impairments along the downlink path. Based on these experiments we may conclude that a transport delay along the communication network of the Fortigate routers can be tolerated up to 250 ms which is achieved in normal situations within Germany.

The reconfiguration tests on the FortiGate router's capability to switch a specified IP-network path within the SD-WAN zone has shown very poor performance results so far. In our preliminary tests more than 3 sec were required to detect a malfunctioning of the working path and to switch over to the secondary one. In our future research more detailed investigations on the network management are required to solve the related availability issues.

4 Conclusions

Nowadays, new advanced Internet-of-Things applications are enabled by high-speed communication and cloud computing technologies. The latter can support the effective processing of smart e-health services such as telesurgery, cf. [1,8]. In our paper we consider the system architecture of a classical telesurgery system. We have discussed the extension of its functional modules to support remote surgery based on a reliable multi-path communication among its components. We have also provided an investigation of important key performance parameters of such remote-controlled telesurgery services. To achieve deeper technical insights on the system's performance of such a distributed architecture and its inherent communication flows, a telesurgery prototype with its major control components that are inspired by the well-known Artis system has been emulated within a virtualized test bed by several GNS3-experiments. In this way we have investigated fundamental computing and network performance indices of our prototype.

Based on these experiments we may conclude that a transport delay within the incorporated communication network of the prototypical telesurgery infrastructure can be tolerated up to 250 ms. Assuming a symmetric transfer delay, we conclude that a maximal transport delay of 125 ms in each direction can be tolerated to assure a proper remote-controlled telesurgery service in our setting. We have also illustrated that a multi-path concept has to be employed to guarantee a high availability level of the remote-controlled telesurgery service.

Furthermore, these GNS3-experiments illustrate that an intensive training of remotely operating physicians is needed to handle such a complex telesurgery infrastructure. Moveover, additional measurement-based QoE-studies are needed to elaborate on those complex treatment scenarios. The latter insight can indicate some feasible ways to realize advanced system architectures including a secure distributed data processing and communication architecture based on 5G/6G technology and its service category ultra-reliable low latency (URLL).

Acknowledgment. This feasibility study was done while Mr. Gerwien was working as master student in the Computer Networks group at the University of Bamberg. The other authors are very much indebted to his efforts in implementing a first virtualized prototype of the telesurgery architecture by means of the GNS3 emulator.
Furthermore, the authors are indebted to the staff of Siemens Healthineers AG, Germany, who provided on request substantial technical insights for the described research on the design and operation of modern telesurgery systems.

References

1. Al-Fuqaha, A., et al.: Internet of Things: a survey on enabling technologies, protocols, and applications. IEEE Commun. Surv. Tutor. **17**(4), 2347–2376 (2015)
2. Barba, P., et al.: Remote telesurgery in humans: a systematic review. Surg. Endosc. **36**, 2771–777 (2022)
3. Cech, H.L., Großmann, M., Krieger, U.R.: A fog computing architecture to share sensor data by means of blockchain functionality. In: IEEE International Conference on Fog Computing (ICFC) 2019, pp. 31–40 (2019)
4. Fortinet: Fortinet - Next-Generation Firewall (2024). https://www.fortinet.com/products/next-generation-firewall
5. GNS3: Getting Started with GNS3 (2024). https://docs.gns3.com/docs/
6. Hoßfeld, T., et al.: QoE of YouTube Video Streaming for Current Internet Transport Protocols. In: Fischbach, K., Krieger, U.R. (eds.) MMB & DFT 2014. LNCS, vol. 8376, pp. 136–150. Springer, Heidelberg (2014)
7. Internet Engineering Task Force: Security Architecture for the Internet Protocol (2024). https://datatracker.ietf.org/doc/html/rfc2401
8. Islam, S.M.R., Kwak, D., Kabir, M.H., Hossain, M., Kwak, K.S.: The Internet of Things for health care: a comprehensive survey. IEEE Access **3**, 678–708 (2015)
9. Kreutz, D., et al.: Software-defined networking: a comprehensive survey. Proc. IEEE **103**(1), 14–76 (2015)
10. Legeza, P., et al.: Impact of network performance on remote robotic-assisted endovascular interventions in porcine model. J. Rob. Surg. **16**(1), 29–35 (2022)
11. Linux man-pages project: tc(8) - Linux manual page (2024). https://man7.org/linux/man-pages/man8/tc.8.htmll
12. Madder, R.D., et al.: Robotic telestenting performance in transcontinental and regional pre-clinical models. Catheter. Cardiovasc. Interv. **97**(3), E327–E332 (2021)
13. Madder, R.D., et al.: Network latency and long distance robotic telestenting: exploring the potential impact of network delays on telestenting performance. Catheter. Cardiovasc. Interv. **95**(5), 914–919 (2020)
14. McGhin, T., Choo, K.-K.R., Liu, C.Z., He, D.: Blockchain in healthcare applications: research challenges and opportunities. J. Netw. Comput. Appl. **135**, 62–75 (2019)
15. Navarro, E.M., et al.: A new telesurgery generation supported by 5G technology: benefits and future trends. Procedia Comput. Sci. **200**, 31–38 (2022)
16. Penn, J.W., et al.: Fifth generation cellular networks and neurosurgery: a narrative review. World Neurosurg. **156**, 96–102 (2021)
17. Philips: Philips announces new augmented-reality surgical navigation technology designed for image-guided spine, cranial and trauma surgery. https://www.philips.com/a-w/about/news/archive/standard/news/press/2017/%20170112-philips-announces-new-augmented-reality-surgical-navigation-technology-designed-for-image-guided-spine-cranial-and-trauma-surgery.html. Accessed 12 Jan (2017)
18. Siemens Healthcare AG. Artis - Systems for interventional angiography (2024). https://www.siemens-healthineers.com/de/angio/artis-interventional-angiography-systems
19. Siemens Healthineers AG. ARTIS icono (2024). https://www.siemens-healthineers.com/de/angio/artis-interventional-angiography-systems/artis-icono
20. Thompson, J.M., et al.: Human factors in telesurgery: effects of time delay and asynchrony in video and control feedback with local manipulative assistance. Telemed. J. **5**(2), 127–221 (1999)

21. Uhl, T., et al.: H.264/AVC and H.265/HEVC codecs in the IP environment: a comparison study of QoE, QoS and UX. In : 2021 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), pp. 1–6 (2021)
22. VMWare (2024). https://www.vmware.com/