An Investigation of the Network Characteristics and Requirements of 3D Environmental Digital Twins for Inspection Robots

Document Version
Accepted author manuscript

Citation for published version (APA):
Kivrak, H., Baniqued, P. D., Watson, S., & Lennox, B. (2022). *An Investigation of the Network Characteristics and Requirements of 3D Environmental Digital Twins for Inspection Robots*.

An Investigation of the Network Characteristics and Requirements of 3D Environmental Digital Twins for Inspection Robots

Hasan Kivrak, Paul Dominick E. Baniqued, Simon Watson and Barry Lennox
Department of Electrical and Electronic Engineering
The University of Manchester
Manchester, United Kingdom
{hasan.kivrak, paul.baniqued, simon.watson, barry.lennox}@manchester.ac.uk

***Abstract*—Digital twins tend to be on the way of becoming the future of robots, artificial intelligence, and IoT devices, especially in industrial applications. Creating a digital twin solution will help to solve challenges faced in managing the tasks in their operating environment since it enables an integrated solution for the framework that seamlessly connects to their physical counterparts with the latest internet technologies. In this study, we aim to develop a synchronous, situational-aware, bi-directional/multi-directional digital twin platform that allows us to perceive real-time/simultaneous flow of sensory environmental data and remotely operate robots through digital twins for continuous inspection. To achieve this aim, we will investigate the interoperability issues in robot teleoperation with 3D mapping of a remote unknown environment case scenario.**

***Index Terms*—environmental digital twins, cyber-physical systems, network requirements, inspection, maintenance, repair robots, cloud robotics**

I. INTRODUCTION

Digital twins have become essential tools in developing novel features for industrial robots such as those applied in remote inspection, maintenance and repair. They allow us to foster innovation at a pace faster than traditional methods [1]. A digital twin is a real-time virtual representation of a physical object or process. Primarily used in simulations and data visualisations during the design and development stage, digital twins have since expanded their scope as vital components of a cyber-physical system [2] [3]. When applied to industrial robots, this includes a digital twin instance of its mechanical and electronic components. However, digital twins can also refer to digital environments based on actual locations where a robot may be operated [4]. The purpose of environmental digital twins in robotics is to digitally represent and recreate the physical boundaries and external processes that interact with the robotic device. In this way, a robot and its intended user are provided with accurate information on its surroundings, leading to better and more efficient mission planning.

A robot must rely on its perception and mapping capabilities to represent environmental digital twins accurately. This process depends on how well its sensors record and stream data from the real world to the digital world. In a typical mission, an inspection robot captures data from its environment using various sensors based on its intended purpose (i.e. visual, geometric, temperature, acoustic, etc.). For example, visual sensors use cameras to capture images, while LiDAR sensors and laser scanners use reflected light to render point cloud data and reconstruct a 3D geometric pattern of its immediate environment. At the same time, some sensors are mission-specific such as detecting radiation sources or reading the temperature. These data are then combined with navigational and positional data (e.g. joystick commands, odometry, rotation and acceleration) to generate an accurate representation of the real world.

The capability of inspection robots to be teleoperated from a remote location is vital to its mission. This bilateral communication process requires a large bandwidth of data transmission and a stable connection between the two terminals. While transmission stability can be achieved using tethered (wired) connections between the robot and its controller, these requirements change in long-distance teleoperation scenarios. In such cases, data transmission would rely on wireless internet protocols and cloud server services.

Several attempts to teleoperate robot digital twins via cloud servers have been published in the literature. Lim et al. [5] have successfully established a robot server-client connection by using the robot operating system (ROS) in a virtual private network (VPN). In their study, they calculated an average round trip time of 18.6540 ± 23.7870 ms for a simultaneous localisation and mapping (SLAM) process of a robot digital twin. Pacheco-Gutierrez et al. [6] were able to implement a multiple level-of-detail compression strategy for point cloud data to enable low-latency transmission in a teleoperated robot manipulator digital and physical twin scenario. In a similar study, Aarizou and Berrached [7] reduced 80% of bandwidth consumption and 60% of the transmission time by implementing a discarding strategy for redundant robotic manipulator transforms.

While the cloud-based teleoperation of mobile robots and robot manipulators offer a perfect example of the utility of digital twins, we found an interesting gap in the availability of research studies when it comes to generating environmental digital twins for inspection robots. The network requirements and limitations of streaming simultaneous environmental sensor data in a teleoperation scenario remain unexplored to the best of our knowledge. Furthermore, there is a need to evaluate how network interfaces such as wired and wireless connections affect data transmission metrics such as bandwidth and latency.

This study aims to understand the network requirements and investigate the feasibility of streaming and processing control, perception, and environmental data in a teleoperated 3D mapping scenario using a mobile inspection robot. The network architecture uses a cloud-based system that connects two terminals: (1) a controller and processor terminal based in Manchester, UK and (2) a robot terminal based in Cumbria, UK, with the server instance situated in London, UK. The delivery of joystick commands, odometry data, and point cloud data was used as representative topics and assessed based on their bandwidth, frequency and latency. On the robot end of the system, a comparative analysis of network interfaces was done by performing runs of the experiment either through wired (Ethernet) or wireless (WiFi) modes. On the processor-end of the system, the generation of a 3D voxel mesh based on point cloud data was simultaneously performed to map and visualise an unknown environment.

The results of this study provide a valuable contribution to the development of digital twin environments by benchmarking its requirements for low-latency teleoperation and data transmission. Furthermore, we demonstrate the utility of this setup in the 3D visualisation of unknown environments, a task usually implemented by remote inspection robots. Section II presents the system architecture with focus on network protocols used, while Section III describes its implementation methodology via a case study experiment. We present the results of this experiment in Section IV and, finally, conclude the study in Section V.

II. SYSTEM ARCHITECTURE

A. Robot Operating System (ROS)

ROS, a defacto standard in research robotics, was developed with the idea of offering standardised functionalities to everyone as an open-source middleware that allows ”nodes” (software components) to multi-directionally communicate using a publish/subscribe messaging mechanism. It provides a flexible framework to develop software for teleoperated robotics and distributed computing. The ROS master connects the nodes to one another for peer-to-peer message exchange by using a proprietary protocol over TCP/IP. The ROS Master is also responsible for keeping topic publishers and subscribers in synchronisation.

B. Virtual private network (VPN) based Remote ROS network

Although configuring the ROS network to control nodes through the internet is similar to local access, it has challenges to be solved due to operating in different private networks. Port forwarding [8] can be considered for direct communication, but it may not be possible for all robots that need 4G/5G connections since it is necessary to contact the internet service provider to enable port forwarding. Also, the websockets based Rosbridge protocol [9] can be an alternative approach. However, the websocket server requires a public IP address to be accessed by clients. Benefiting from assigning private IP addresses through a public network, we configured a VPN-based ROS network to allow a secure, encrypted connection with the ROS architecture over the internet as shown in Figure 1.

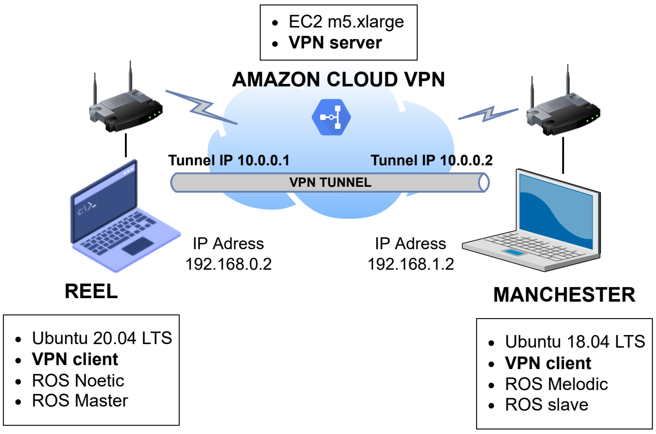


Fig. 1. Cloud based VPN setup

Our approach consists of the design of a centralised architecture for network digital twins, which is facilitated by creating a point in the cloud. The idea is to use cloud-based systems to facilitate multi-user remote operation of the physical twins and incorporate sensory data into digital twins. Amazon Web Services’ Elastic Compute Cloud (Amazon EC2) M5 xlarge instance [10], which is located in London and features up to 10 Gbps of network bandwidth, was used to support ROS infrastructure through the internet between two sites: (1) The University of Manchester (UoM) located in Manchester, UK and (2) the UoM-Robotics for Extreme Environments Laboratory (REEL) located in West Cumbria, UK as shown in Figure 2. Amazon served as the bridge in a configured VPN server, allowing the physical and digital twins to coexist seamlessly.

C. Metrics

The latency, bandwidth and publishing rate, which are commonly used metrics for network performance, were employed to measure and quantify the performance scale and bottlenecks of communication systems in this study. The standard ROS topic Python library, rostopic[1], is used to get the latency and bandwidth information for topics. Network latency is the time it takes for data to move across a network to its intended destination. It is referred to as delay in a network and is usually measured in milliseconds (ms). In order to obtain the latency of the networks, the function “rostopic delay” was used. Once the subscriber receives the message, it calculates the difference between the timestamps of the published and

---

[1] http://wiki.ros.org/rostopic

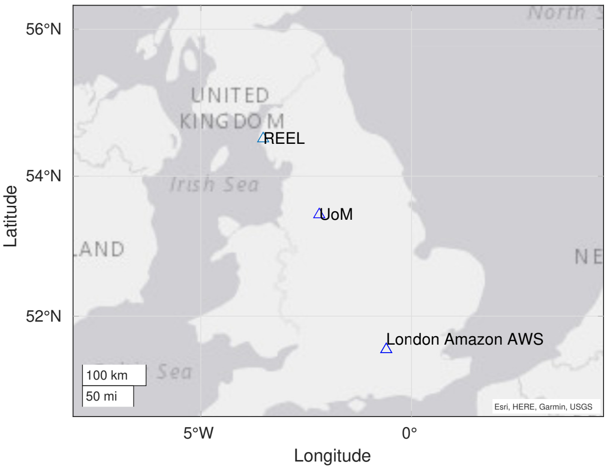


Fig. 2. The location of the Robotics for Extreme Environment Lab (REEL), the University of Manchester (UoM), and Amazon Web Services (AWS) in London.

subscribed messages and returns the float seconds of the topics with min, max, average, and standard deviation. Furthermore, bandwidth (capacity) refers to the amount of data transmitted per second. The “rostopic bw” was used for the monitoring of received network bandwidth consumption by topics and the “rostopic hz” displays the its publishing rate.

III. CASE STUDY

We discuss a simplified teleoperation application with a 3D environmental digital twin case scenario for remote inspection robots in a ROS-based system. We considered the development of environmental 3D digital twin generation through the transmission of 3D lidar (PointCloud2), odometry (Odometry) and motion (Twist) messages. The overall system setup developed in ROS is visualized in Figure 3. A Scout 2.0 robot equipped with a Velodyne-16 sensor and an i7 laptop is built to asses the feasibility of a mobile area in Figure 4(a). In this scenario, the human operator sends velocity commands to move the robot via Joystic Keypad while viewing the 3D representation on a remote computer to build a real-time 3D map of the environment.

A. Scout 2.0 Robot

The Scout 2.0 robot is customised with additional sensors (3D laser range finder and Logitech camera) and processing power, as can be seen in Figure 5. The computer that runs Ubuntu 20.04 LTS with ROS Noetic and is configured as Master is used both for control tasks such as perception and motor control. Robot control board is connected to the computer via CAN-To-USB adapter to expect motor velocities and get feedback from wheel encoders to calculate the odometry.

B. Joystic Keypad

The Logitech F710 wireless joystick has 6 analogue axes and 12 digital buttons and is configured to provide velocity data to the robot. It publishes a /joy message containing the current state of each of the joystick’s buttons and axes. As a safety precaution, one of the buttons is used as an activation button that behaves like an emergency button. Furthermore,

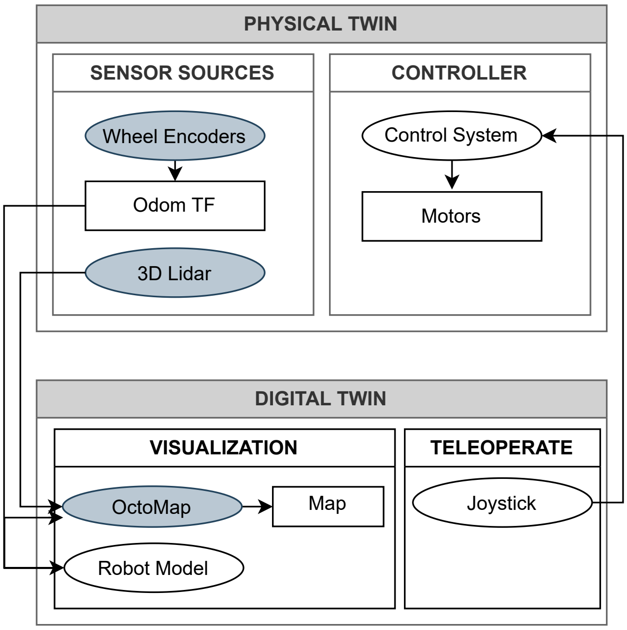
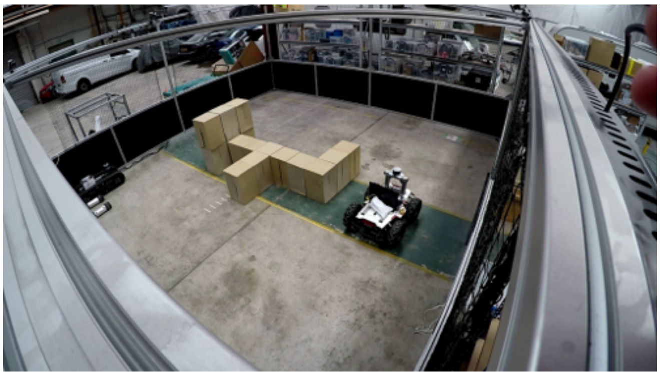
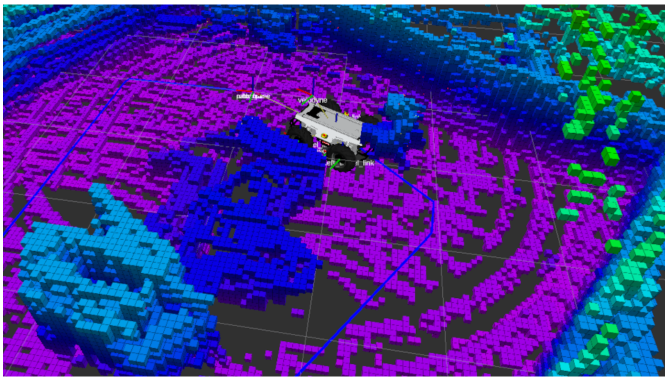


Fig. 3. Overall software system architecture and data flow diagram



(a) Physical Twin



(b) Digital Twin

Fig. 4. Comparison of the physical environment and the generated environmental digital twin

Fig. 5. The Scout 2.0 mobile robot platform provided with modified hardware architecture

four different driving modes in terms of the speed commands are defined to make sure of a driving robot with the intended speed intervals (Figure 6). A local extra remote controller was also ready to take over the control whenever necessary.

Fig. 6. Joystick key-specific functions applied to our system.

IV. RESULTS AND DISCUSSION

This section presents the network performance results obtained from the case study. A video clip demonstrating the results of experiments can be found at the hyperlink[2]. The average bandwidth consumption and of all topics published in the system is presented in Table I. This was calculated as the multiplication of the average message size to the publishing rate of the topic. The internet service upload speed used by the physical robot is measured at 8Mbps (i.e. 1000 kB/s) and 11Mbps (i.e. 1375 kB/s), which is for wireless and wired connections, respectively.

The results in Table I showed that bandwidth consumption of the point cloud topic at a frequency of 10Hz is 5530 KB/s, which is 5.53 and 4.02 times greater than the available bandwidths of both wireless and wired configurations. This leads to a network overhead reduction of 0.603 Hz and a 3.337 s delay on average in the receiving of messages, as can be seen in Table I and Table II. Testing on wireless experiments had higher latencies with higher standard deviations (changing) on all topics compared to wired configurations. Furthermore, Figure 7 shows the average latency performance of network configurations over time for all topics. Although the latencies on point cloud data do not change significantly, the delay in transmitting the data is noticeable and may have an impact on the quality of building a 3D map of the environment.

TABLE I
AVERAGE BANDWITH CONSUMPTION OF THE TOPICS PUBLISHING IN THE NETWORK

Topic name	Published		Received	
	frequency(Hz)	average bandwidth (KB/s)	frequency(Hz)	average bandwidth (KB/s)
point cloud	10	5530 (5.5 MB/s)	0.603	324.32
odometry	50	35.84	49.95	35.80
joy	70	6.09	69.97	6.03

TABLE II
OBSERVED LATENCIES IN THE TELEOPERATION WITH THE 3D MAPPING SCENARIO.

	point cloud (s)		odometry (ms)		joy (ms)	
	Wired	Wireless	Wired	Wireless	Wired	Wireless
Average	3.337	3.000	74.576	101.071	81.904	86.189
Minimum	1.929	2.166	8.18	6.911	38.028	43.056
Maximum	5.625	6.482	365.488	1342.265	229.067	958.311
Std	0.676	0.638	26.332	163.878	29.871	74.050

On the other hand, odometry and joy topic respect bandwidth limits with an average bandwidth consumption of approximately 35 KB/s and 6 KB/s with a frequency of 50 Hz and 70 Hz, respectively. The variance in the decreasing and increasing latencies observed on joy and odometry topics over time might be due to the unstable nature of wireless network connections. Nevertheless, they are unnoticeable and are considered not to have much effect on the remote operation as the latencies are below 100 ms. Also, it is observed in Figure 7 that receiving data is more stable (fewer changes in latencies) in wired experiments.

It is realised that this case study is an example of the overload of ROS topics on the network since the available bandwidth is exceeded in any wired or wireless case. This will result in a problem of not getting back all the data we have on the physical twin to the digital twin and having to expect delays in transmitting and receiving messages. Thus, it may cause inaccuracies between topics that have different published rates and are intended to be evaluated together. To avoid this problem, it is necessary to develop techniques to adjust the publishing rate of topics for bandwidth usage, tuning for viable data transmission in bandwidth-restricted environments or increase network bandwidth.

Moreover, Table III shows how the bandwidth consumption increases in proportion to the number of robots to manage and optimise mixed fleets of robots in a variety of settings at scale.

---

[2] Video of the experiment: https://youtu.be/dE7NjBsrtKk

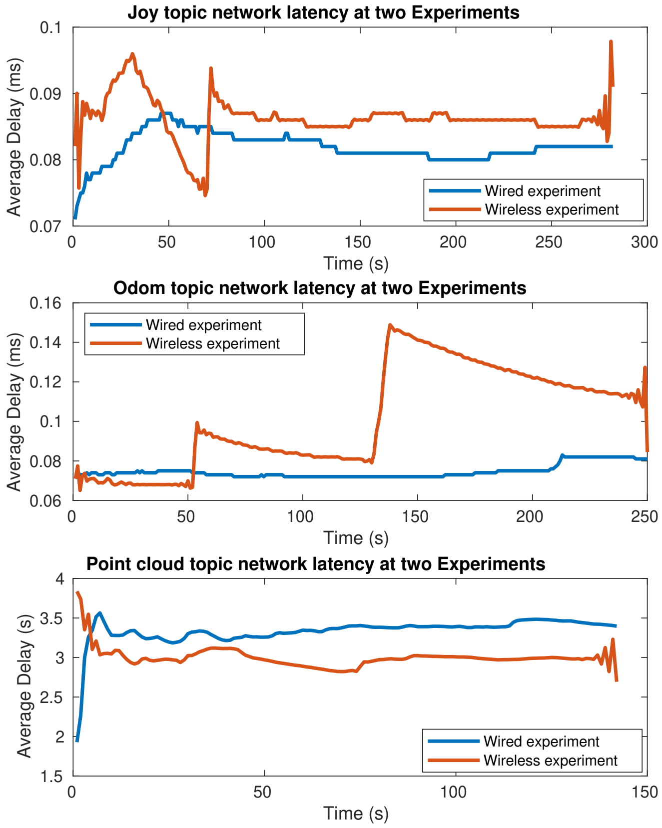


Fig. 7. Average latencies of joy, odometry and point cloud messages over time

TABLE III
BANDWIDTH CONSUMPTION OF FLEETS OF ROBOTS IN A NETWORK THAT PROVIDE ODOMETRY AND 3D LIDAR DATA

Number of robots	Bandwith consumtion (MB/s)	Network speed (Mbps)
1	5.565	44
2	11.1300	89
3	16.6950	132

The combination of odometry and 3D Lidar data provides a 44 Mbps network bandwidth for operating a robot in a 3D environment. As the fleets of robots grow to two and three robots, the network requirements will increase to 89 Mbps and 132 Mbps, respectively.

V. CONCLUSION AND FUTURE WORK

In this study, we investigated the network requirements of real-time streaming and data processing in the teleoperation of a mobile inspection robot to generate environmental digital twins. We have set up a cloud-based VPN to operate a remote ROS network and stream navigation and perception data. We tested the limits of this setup by running a 3D mapping scenario in wired and wireless modes. The current setup provides a preliminary guide to developing future software pipelines that allow a smooth operation and real-time updates of digital twins with real-world data for continuous remote inspection.

In future work, we will extend our approach by using other platforms such as ROS-Unity3D based systems and developing data visualisation methods for incorporating sensory data into the digital twin. The developed approach can also be applied to new and larger scenarios with more robots by offloading computationally-heavy processes to the off-the-shelf Amazon webserver to mitigate the challenges associated with scaling up with fleets of robots.

ACKNOWLEDGMENT

This work was funded and supported by the UK Research and Innovation (UKRI funded project EP/P01366X/1).

REFERENCES

[1] B. David, B. Rob, C. Paul, H. Sabine, K. James, L. David, N. Paul, R. Rob, S. Julia, and W. Elaine, "The cyber-physical infrastructure," 2022, last accessed 22 February 2022. [Online]. Available: https://www.gov.uk/government/publications/cyber-physical-infrastructure

[2] E. G. Kaigom and J. Roßmann, "Value-driven robotic digital twins in cyber–physical applications," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3609–3619, 2021.

[3] J. Douthwaite, B. Lesage, M. Gleirscher, R. Calinescu, J. M. Aitken, R. Alexander, and J. Law, "A modular digital twinning framework for safety assurance of collaborative robotics," *Frontiers in Robotics and AI*, vol. 8, 2021. [Online]. Available: https://www.frontiersin.org/article/10.3389/frobt.2021.758099

[4] G. S. Blair, "Digital twins of the natural environment," *Patterns*, vol. 2, no. 10, p. 100359, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S266638992100221X

[5] J. Z. Lim and D. W.-K. Ng, "Cloud based implementation of ros through vpn," in *2019 7th International Conference on Smart Computing & Communications (ICSCC)*. IEEE, 2019, pp. 1–5.

[6] S. Pacheco-Gutierrez, H. Niu, I. Caliskanelli, and R. Skilton, "A multiple level-of-detail 3d data transmission approach for low-latency remote visualisation in teleoperation tasks," *Robotics*, vol. 10, no. 3, 2021. [Online]. Available: https://www.mdpi.com/2218-6581/10/3/89

[7] M. L. Aarizou and N. Berrached, "Ros-based telerobotic application for transmitting high-bandwidth kinematic data over a limited network," *International Journal of Control, Automation and Systems*, vol. 17, pp. 445–453, 2018.

[8] S. S. H. Hajjaj and K. S. M. Sahari, "Establishing remote networks for ros applications via port forwarding: A detailed tutorial," *International Journal of Advanced Robotic Systems*, vol. 14, no. 3, p. 1729881417703355, 2017.

[9] C. Crick, G. Jay, S. Osentoski, B. Pitzer, and O. C. Jenkins, "Rosbridge: Ros for non-ros users," in *Robotics Research*. Springer, 2017, pp. 493–504.

[10] I. Amazon Web Services, "Amazon ec2 m5 instances," 2022, last accessed 28 February 2022. [Online]. Available: https://aws.amazon.com/ec2/instance-types/m5/