Pertanika J. Sci. & Technol. 32 (4): 1875 - 1890 (2024)

PERTANIKA JOURNALS

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Simulating Spanning Tree Protocols in a Cable-based Tsunameter System with an Arbitrary Number of Ocean Bottom Units

Mohammad Hamdani*, Anak Agung Ngurah Ananda Kusuma, Dedy Irawan, Tahar Agastani and Xerandy

Research Center for Telecommunications, National Research and Innovation Agency, KST BJ Habibie, Tangerang Selatan, Banten 15314, Indonesia

ABSTRACT

As a country with the fourth largest population in the world prone to tsunami disasters, Indonesia needs a reliable, timely early warning system to mitigate the impact of disasters. Indonesia cable-based tsunameter (INA-CBT) is an undersea tsunami detection system comprising undersea pressure sensors and a shore station connected by underwater fiber optics designed to provide early warning to the threatened area. Since this system performs a critical role in disaster mitigation, the system must be resilient to link failure and deliver timely warning information. This system is still in its early implementation and still on a small scale. Network-wise, it uses a proprietary Layer 2 (L2) communication protocol. Extending such a network to a larger scale and assessing the system's performance may introduce challenges due to high costs and offer less flexibility. This paper aims to address those challenges and presents a scalable simulation framework of the INA-CBT system by using L2 open protocols such as spanning tree protocol (STP) and rapid spanning tree protocol (RSTP). The framework is conducted in OMNET++ simulator. The experiment shows that the downtime duration using STP and RSTP is still below the allowed value. RSTP shows a faster failover time than STP, but RSTP downtime duration fluctuates compared to a steady one of STP. The experiments also demonstrated that the variation of downtime is affected by two aspects: the number of ocean bottom units (OBUs) in the network and the position of their blocked port.

Keywords: Failover time, INA-CBT, network reliability, OMNET++, spanning tree protocols

ARTICLE INFO

Article history:
Received: 10 May 2023
Accepted: 17 January 2024
Published: 25 July 2024

DOI: https://doi.org/10.47836/pjst.32.4.22

E-mail addresses:
moha049@brin.go.id (Mohammad Hamdani)
anak001@brin.go.id (Anak Agung Ngurah Ananda Kusuma)
dedy012@brin.go.id (Dedy Irawan)
taha001@brin.go.id (Tahar Agastani)
xera001@brin.go.id (Xerandy)
* Corresponding author

ISSN: 0128-7680
e-ISSN: 2231-8526

INTRODUCTION

A reliable and timely early warning system is essential in a nation's disaster management system to mitigate the adverse impacts of natural disasters. Japan, for example, has established its national marine science and technology agency, the Japan Agency for Marine-Earth Science and Technology (JAMSTEC), to promote ocean research and activities to better understand the ocean and earth. JAMSTEC also encourages post-disaster evaluation and research to manage, reduce, and mitigate disaster risks (Mikada et al., 2003). National Oceanic and Atmospheric Administration (NOAA) is a United States agency that deals with global ocean, atmospheric, and climate research and monitoring, which especially holds a crucial role during the nation's disaster season (Domenikiotis et al., 2003). Indonesia, a nation that is prone to tsunami hazards due to its unique location's geology characteristics, has developed a national early tsunami warning system called the Indonesia Tsunami Early Warning System (INA-TEWS), which was strongly motivated by devastative experience after being struck by an unprecedented tsunami in December 2004. One major component of INA-TEWS infrastructure is an underwater cable-based tsunami detection system, which is referred to as INA-CBT. This system monitors sea conditions for possible tsunami occurrence, especially following a strong ocean-epicenter earthquake event (Privadi et al., 2021).

The INA-CBT system architecture is a series of sensors connected by fiber optic cables stretched on the seafloor, linking the sensors to the shore landing station (LS). The observed sea points have the potential to detect earth cracks based on previous research (Widiyantoro et al., 2020). Those sensors are accelerometers, pressure sensors, and thermos sensors. They actively monitor and collect underwater conditions data for earthquakes, sea levels, and potential tsunamis. These data are sent to LS and then to the data center for consolidation and further processing. This results in intense data transfer between sensors and servers in the data center.

Incoming and outgoing data traffic across the sensors are only managed at L2 layer switching due to the complexity of using various devices. Because each sensor is embedded in an OBU with limited space and dimension, the devices must be efficient and have a small dimension to avoid overheating from their electronic components.

The switching process must be made fast during normal data transmission events, and quick self-recovery must be performed upon failure due to a link loss incident; this might change the switching path. According to the tsunami detection algorithm implemented in existing INA-CBT systems (Iqbal et al., 2021), the maximum downtime for such application is set at most six minutes to prevent failure of the tsunami detection system in detecting anomalies or incidents that might have occurred.

INA-CBT infrastructures were deployed in two different locations: Labuan Bajo (Purwoadi et al., 2023) and Rokatenda (Iqbal et al., 2021). Both implement ring topologies

with a proprietary, single-brand dependency ring protocol. These topologies are still feasible for both infrastructures since only a limited number of OBUs were observed, less than three units. At this configuration, the distance between the farthest sensor and the LS can still be accommodated by using an optical signal, even at the expense of using extra optical fiber lines. Exercising the system performance while incorporating proprietary protocol is still manageable at this scale. However, design challenges come into play when the network size of the system increases. Simple star topology may not be able to facilitate a larger number of OBUs, and multi-hop optical links are required. The ring topology must also be augmented on a larger scale and have more complex configurations. A proprietary L2 protocol with more complex configurations may limit design flexibility and compound performance assessment efforts. Furthermore, using proprietary L2 protocols may incur high implementation costs for larger systems.

A simulation framework is needed to address the issues above and evaluate the network performance at a scale. Hamdani et al. (2023) discussed that our study of the existing system proved that open protocols, i.e., STP and RSTP, could perform well under a stringent downtime requirement. The simulation demonstrated that the system suffered from downtime as long as 50 s by using RSTP, while it took only 22 s downtime before recovering by using RSTP. These results were far below the six minutes of maximum downtime allowed by the INA-CBT tsunami detection algorithm (Iqbal et al., 2021). This result also opened an opportunity for STP and RSTP protocol as a potential substitution for existing INA-CBT networks (Hamdani et al., 2023), even though they were still operated on a simple topology, namely one LS and two OBUs.

The future development of INA-CBT requires more OBUs to reach longer distances and better data accuracy from more sensors. The data communication will be even more complex, so extensive testing is needed before implementation. Our contribution to this paper is to test with a simulator the INA-CBT designs with a single LS and an arbitrary number of OBUs. Several simulations were carried out to measure the downtime that may occur in every possible incident and to assess whether the downtime is still within the timeframe required by the INA-CBT system.

MATERIALS AND METHODS

Related Work

STP is a link management protocol that creates logical connections between L2 nodes. The spanning tree protocol was first proposed in 1985 by Digital Equipment Company (DEC) (Perlman, 1985). In 1990, the Institute of Electrical and Electronics Engineers (IEEE) published the first standard for the protocol as IEEE 802.1D based on an algorithm designed by Perlman (2000). IEEE 802.1D standard, known as STP, has been widely utilized to eliminate traffic loops that might trigger broadcast storms in a network. In 2001, the IEEE

improved the STP protocol, namely RSTP, by releasing the IEEE 802.1w standard, followed by developing multiple spanning tree protocol (MSTP), entitled the IEEE 802.1s standard. RSTP and MSTP are two newer spanning tree protocols, where the improvement is in the speed of convergence and the capacity to handle multiple spanning trees with virtual local area networks (VLANs), respectively.

All STP models will prevent network loops by blocking one of the ports. The blocking port is chosen based on the device's lowest bridge ID, where the bridge ID is calculated from the bridge priority, the system ID extension/VLAN, and the bridge's MAC address. The formula for a bridge ID is in Equations 1 and 2:

$$bridgeID(8Bytes) = bridgepriority(2Bytes) + MACAddress(6Bytes) \quad [1]$$

$$bridgeID(8Bytes) = bridgepriority(4bits) + systemID\ extension/VLAN\ (12bits) + MACAddress(6Bytes) \quad [2]$$

The device with the lowest bridge ID will be elected as a root bridge, a reference point for all interconnected switches in a spanning tree topology. As shown in Figure 1, Switch1 (SW1) is elected as a root bridge because its MAC address is the lowest one, and bridge priorities are the same with the default value (32768) in all switches. Note that the election can be manipulated by setting a lower bridge priority value than the default on the desired switch. All ports that forward traffic and not facing the root bridge are denoted as designated ports (DP), while the ports that face a root bridge are denoted as root ports (RP). Ports that cannot forward traffic are denoted as blocked ports (BP).

A spanning tree can be used for redundancy and loop avoidance in a submarine communication system (Premod et al., 2013). Spanning trees can also improve redundancy in undersea sonar communication systems. In addition, the failover times of the three suggested redundancy approaches, namely STP, RSTP, and link aggregation (LAG), were compared, respectively. Similarly, in industrial network redundancy, as demonstrated by Gunnar's research, the utilization of the RSTP is prevalent, particularly in ring topologies (Prytz, 2007). The redundancy selection is determined based on the system's needs. For example, if the system serves a critical role and needs to be responsive, it is necessary to consider protocol with the fastest failover.

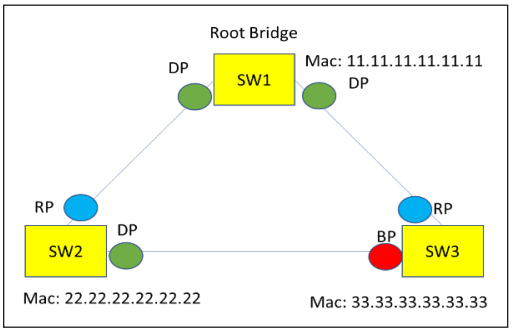


Figure 1. Spanning tree port assignment

The application of spanning trees is also applied in network planning (Wang et al.,

2021). It formulated integer linear programming (ILP) to optimize the design of an optical network. The ILP method could achieve an optimal design with lower latency without network loops. Network design does not always consider Layer 2 communication between nodes. It can use Layer 3 instead, but in some cases, it requires Layer 2 communication due to data switching speed or the limited environment of the system.

In the context of improving network resiliency, the study by Marchese and Mongelli (2012) revealed that the spanning tree protocol has been the most cost-effective scheme among other L2 protocols needed to increase the resiliency of a network against link failures, especially in a ring topology. Spanning trees also build a resilient power grid topology (Kaiser & Witthaut, 2021). While using the spanning tree algorithm, calculations for link failures are conducted to analyze the topological structure to avoid the spread of link failures. It is proven that the spanning tree-based algorithm can increase the resilience of the network being built.

In general, spanning trees have been widely used to build network resilience because they are simple and easy to implement. Moreover, switch devices spanning three features are easy to find on the market. In the context of INA-CBT, in addition to network resilience among sensors and the LS, one must ensure sensor data reach the read-down station (RDS) within acceptable latencies constrained on the allocated bandwidth (Kusuma et al., 2022).

The STP algorithm is also widely used in advanced networks, such as virtual private networks (VPN) and software-defined networks (SDN). The work of Kolarov et al. (2004) discussed traffic engineering design for networks that support global open ethernet (GOE) architecture (Iwata et al., 2004), which was designed for VPN over optical network infrastructure. While the existing extensions of STP, such as RSTP and MSTP, use link metrics that are inversely proportional to link capacity, this work took the current link load into account and formulated a function to calculate the link cost. This scheme is applied in various topologies under different scenarios. Kolarov et al. (2004) observed significant improvement in network load balancing by having link cost as a function of the link load.

While STP protocol and its variants are utilized vastly in terrestrial wired and wireless networks, their application in underwater wireless networks may be arguably insignificant since this type of network in real practice is infrequent and mostly implemented wirelessly. This condition is probably due to the harsh water environment and being developed for a few specific purposes. Nevertheless, some works studied the application of the protocol in underwater networks.

Zhang et al. (2008) investigated the effectiveness of the spanning tree protocol in increasing the performance of underwater 3-D geographic routing in a wireless underwater sensor network. They explained that most spanning tree algorithms operate on a centralized and top-down approach, which results in poor routing performance for such networks. Therefore, they devised a spanning tree algorithm that used a bottom-up fashion with the

traffic load in mind called a traffic-aware routing tree. Concerning the context of this paper, it is a potential improvement applicable to our CBT network and worth further investigation.

Rather than directly implementing the current spanning tree protocol, Mupparapu et al. (2005) took the core idea of the spanning tree algorithm. It developed a novel routing scheme for underwater mobile ad hoc networks called AUSNET and COFSNET. This algorithm was developed based on dynamic source routing (Johnson & Maltz, 1996). This scheme used a spanning tree algorithm to compute the shortest path to the destination to reduce overhead in the route discovery process.

Proposed Design and Method

This work proposed a CBT infrastructure with a single LS and multiple OBUs (Figure 2). These OBUs are arranged in series and installed on a straight line of sub-marine optical fiber. Although the OBUs are placed in such a way, they are logically connected in a ring fashion, and a spanning tree protocol runs on top of this topology. The ring topology is fairly efficient when operating under limited available transmission media (optical fiber) by long-distance, single-ended LS.

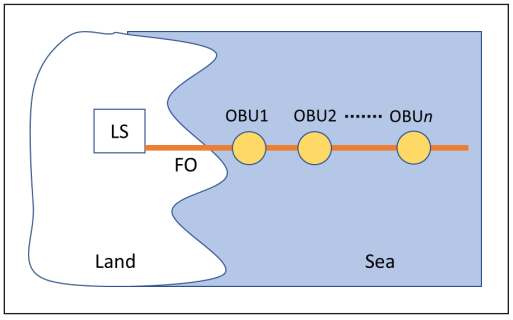


Figure 2. INA-CBT topology with a single landing station (LS)

As shown in Figure 2, the submarine-grade fiber optic cable extends from the LS onshore towards the sea. At a pre-determined distance interval, one or more OBUs are installed and linked to the optical fiber, in which the number of OBUs varies depending on needs, marked with OBU1, OBU2, and OBUn. The distance is not necessarily to be even. Each OBU has sensors to detect sea conditions and predict possible tsunamis, especially following sea-originated strong earthquake events.

Typically, a submarine fiber optic connects two communication endpoints separated by the sea. However, in a single LS INA-CBT configuration, the cable is only terminated at one side of the communication endpoint, the LS. Since LS also provides power to the OBUs, the electrical power is supplied from one source at the LS. So, the problem of breaking one of the connections is crucial because it can cause a loss of communication to a more distant OBU point.

This paper proposes a ring topology design such that the linkage from LS to OBUs forms a loop. This design can increase the robustness of the network in case of connection loss at one of the OBUs or fiber optic cores that connect between OBUs.

As depicted in Figure 3, data communication is designed to be connected sequentially from LS to OBU1, OBU2, OBU3, and OBUn. Each node represents a switch, which then

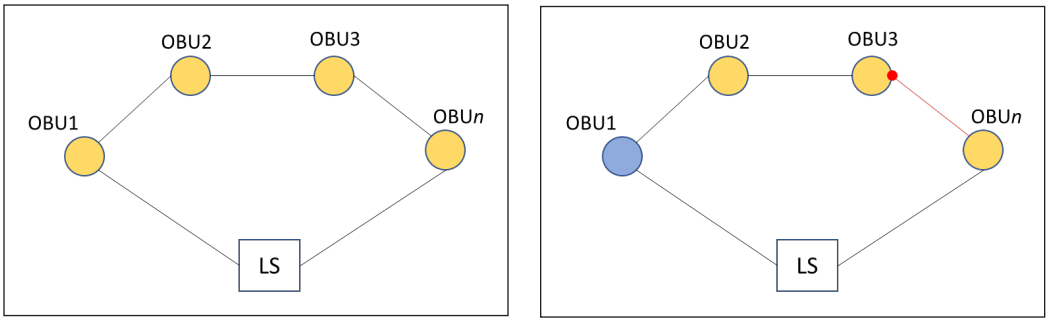


Figure 3. (a) Ring topology of INA-CBT before spanning tree being formed; and (b) Ring topology of INA-CBT after spanning tree has formed

forms an L2 loop. However, STP will avoid that by blocking one of the interconnection ports between two nodes in a ring topology. The blocked port that occurs can be located anywhere between two nodes in the network, depending on the position of the root bridge.

LS is the data destination transit center of all connected OBUs; hence, the root bridge must be at OBU1, and the active path (AP) will be OBU1 to LS. This condition can be done by configuring the OBU1 switch to become the root bridge using the bridge priority parameter with the smallest value. So, the blocked port will be among the last OBUs, and the OBU, which is at the end of the sequence (after the blocked port), will use a different path as an AP. If there is a communication loss on one of the links other than the blocked link, OBU will change its data route to the backup path (BP) depending on the location of the loss. The last block port will change its state to forward so that it can forward data, and the network loop will not happen since one of the links on the network has been broken.

In this work, ten OBUs are used as the maximum number of units and arranged in series. This arrangement considers the optical transmission capability for long-haul communications. The number of OBUs analyzed in this work starts from three since it extends the previous work (Hamdani et al., 2023), which used two OBUs in a series.

Each topology with a different composition of the number of OBU also has different characteristics. While the root bridge is set to OBU1 (since it is the nearest OBU with LS), the placement of the blocked port depends on the total number of nodes (LS is included on the total number of nodes) on the ring topology (odd or even). If the total number of nodes is even, it is obvious that the blocked port is in a certain OBU. For example, Figure 4(a) shows 6 nodes (5 OBUs and 1 LS) with the blocked port in OBU4. From the root bridge (OBU1) point of view, it counts the number of links in each tree (left or right) side. Then, a particular port that becomes a blocked state depends on the bridge ID calculated as Equation 1. If the bridge ID to the left tree is lower than the right one, the blocked port will face the right tree (OBU4 facing OBU5).

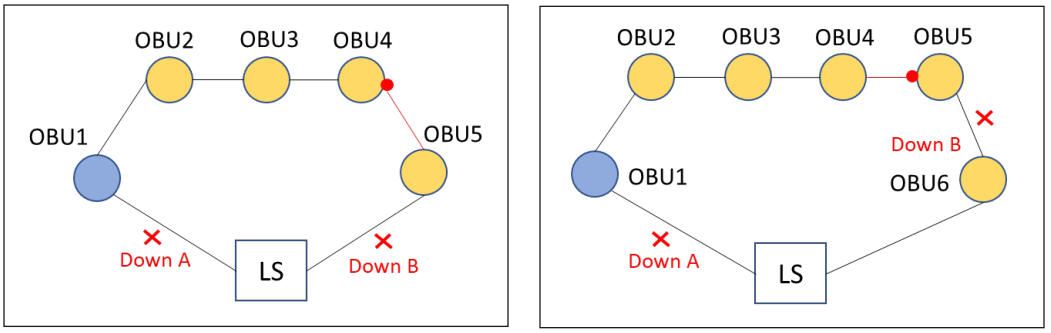


Figure 4. (a) Ring topology with single LS and five OBUs; and (b) Ring topology with single LS and six OBUs

If the number of nodes is odd, the blocked port will be between two OBUs, as shown in Figure 4(b), since the root bridge will balance all links connected from each tree side. Figure 4(b) shows 7 nodes (6 OBUs and 1 LS) where the blocked port is either in OBU4 or OBU5. Then, the calculation of bridge ID starts to determine which port will be assigned a blocked state. In this example, the port of OBU5 facing OBU4 is elected as a blocked port.

The down simulation is carried out on two links: the closest link between the root bridge to LS (down A) and the closest link to the blocked link to LS in the opposite direction (down B). This simulation is considered the worst incident that may occur during downtime. For example, when the nearest link is broken between the root bridge to the LS, the event being explored is when this broken link causes a major impact on the spanning tree topology. While the nearest link breaks between the blocked link to the LS in the opposite direction from the AP root bridge, the event being explored is the incident that causes an impact on any communication and the duration of the downtime.

Downtime measurement is performed on two events: failover and failback. Failover is downtime that occurs during down incidents until data traffic returns to normal. On the other hand, failback is downtime after the previous down incident has been recovered. It is known that the spanning tree checks its topology so that if there is a change, there will be downtime consequences under certain conditions. In the simulation, failover occurs at 100 seconds and failback at 300 seconds.

This simulation also compares using two spanning tree protocols, STP and RSTP. The two protocols are commonly used, and it is easy to get hardware devices supporting these protocols. Thus, simulation results will be an appropriate reference for real implementation. The method used is that each OBU performs a probing ping to the LS every second, with different time intervals, to avoid collisions. Downtime observations are made at each OBU when failover and failback occur from each down simulation. Furthermore, the downtime impact is compared to when the network uses STP or RSTP.

Simulation

A simulation is needed to test the availability level based on a scenario where communication is lost on one of the links. Simulation can be done using software completely or by building a testbed system that represents the system being built (Babu & Kumar, 2022). Both simulation techniques have their advantages and disadvantages. For example, a simulation using software completely has advantages in the ease of changing parameters and values of a component, the scalability for testing various scenarios and designs, and the cost to implement. However, the drawback is that the user cannot experience hands-on in configuring physical devices. On the contrary, using testbeds offers hands-on experience with the real system. The drawbacks of the testbed system are that it is difficult to modify parameters and values as desired, the implementation cost is expensive, and its impact needs to be more scalable.

This work used Omnet++ 6.0 with the INET 4.4 framework to simulate a full software simulation. Omnet++ is a discrete event network simulator commonly used in academics because many network component parameters can be set and even developed for experimental needs (Varga, 2010). Omnet++ is easy to install on Windows and Linux platforms and has a friendly graphical user interface (GUI). The INET4.4 framework is a collection of features and functions that can be used in ethernet-based network simulations (Steinbach et al., 2011) so that users can easily create scenarios using various available protocols, such as Transmission Control Protocol (TCP) and User Datagram Protocol (UDP), even with varied traffic loads. In this simulation, the INET4.4 framework was used because it is in accordance with the INA-CBT communication scenarios, where each sensor inside the OBU communicates using TCP to LS. The available modules in the framework can represent all simulation components and the modules utilized in this simulation are shown in Table 1.

The first step in using the Omnet++ simulator is to build a network topology with the NED file, which functions to define all the components used in the simulation and create the topology (Figure 5). The next step is creating an INI file that defines simulation parameters, such as IP addresses, traffic applications, data transmission time intervals, downtime scenarios, and the simulation workflow (Algorithm 1).

Table 1
Modules that are used in the simulation

Module	Objective
StandardHost	To represent the sensor inside the OBU for sending traffic data
EtherSwitch	To represent the switch inside the OBU for creating interconnection
Scenario Manager	To define the scenario, i.e., disconnection and re-connection link
L2NetworkConfigurator	To create a layer two topology with a spanning tree protocol
Ipv4NetworkConfigurator	To configure IPv4 and static routing

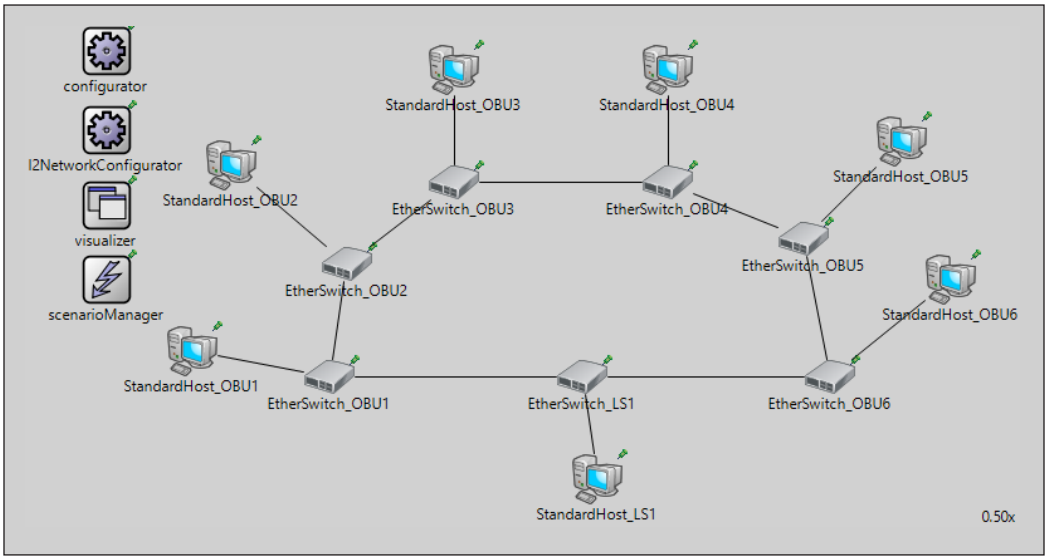


Figure 5. Simulation topology on Omnet++

Algorithm 1

INA-CBT Spanning Tree Simulation

1	*S_n*	← *switch of n node (for LS or OBUn)*
2	*H_n*	← *host of n node (for LS or OBUn)*
3	*con_n*	← *down connection of n (for A and B scenarios)*
4	*i_n*	← *interval of a ping probe*
5	*tf*	← *time failure (100 s)*
6	*tb*	← *time failback (300 s)*
7	***Input:***	
8		*set topology based on S_n*
9		*set ip_address on each host*
10		*set link between S_LS to S_OBU1 to be the main_path*
11		*set protocol STP or RSTP on switches*
12	***Process:***	
13		*set bridge_priority S_OBU1 to the lowest value (4096)*
14		*set bridge_priority S_LS to the second_lowest (8192)*
15		***Loop***
16		*probe ping from all H_OBUn to H_LS with i_n*
17		***If*** *time meet to tf*
18		*scenario down con_n starts*
19		*some H_OBU will be down*

wait time to recover the success of sending a ping
Else *time meet to tb*
scenario recover con_n starts
some H_OBU will be down
wait time to recover the success of sending a ping
End loop *if the spanning tree has been stable*
Output:
Downtime

RESULTS

A downtime was measured based on the ping probe made by each OBU to LS, and if it does not get a reply until a certain time, it is set as a timeout. Furthermore, downtime was tested using several criteria: failover and failback scenarios, the use of STP and RSTP, and simulated downtime on links A and B.

Since measurements were taken for each OBU, and several OBUs have the same downtime pattern, we summarize the OBU categories into nearest, middle, and farthest. The nearest OBU is closest to the down point (link A or link B); for example, down on link A (in all topologies), the nearest OBU is OBU1. The farthest is the OBU that is the farthest from the down point and the closest to the blocked link from the spanning tree calculation; for example, down on link A in a topology with six OBUs, the farthest OBU is OBU5. Meanwhile, the middle is a measurable OBU between the nearest and farthest, with the worst downtime value among them, regardless of the odd or even number of OBUs involved.

Link Down at A

From the down simulation on link A, failover and failback using STP have the same pattern starting from four OBUs. Figures 6 and 7 show that the maximum downtime is 121 s at the nearest OBU and middle OBU. Whereas for three OBUs, the downtime on failback was 31 s faster than the downtime on failover.

Using RSTP, the downtime for the same scenario (link A down) is shorter than that of STP. In the failover simulation, the downtime values fluctuate for the nearest OBU and middle OBU (Figure 8), with a higher value for an odd number of OBUs, with a difference of 2 s longer between the nearest and middle. At the farthest OBU, the downtime values are steady at 16 s.

Figure 9 shows the failback downtime results of link A down with RSTP. The result shows that the nearest OBU's downtime values are steady at 1 s, the farthest OBU's downtime values surge starting from five OBUs and then steady at 27 s, and the middle OBU's downtime values tend to be linear with an increase in the number of OBUs.

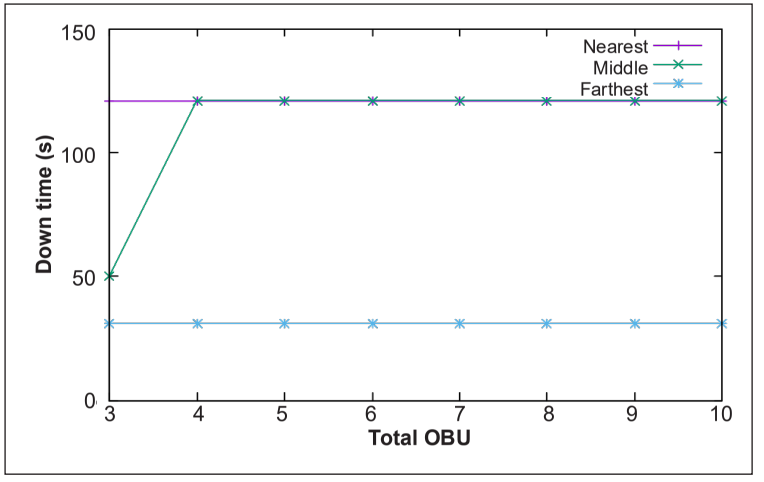


Figure 6. Failover downtime of link A down, using STP

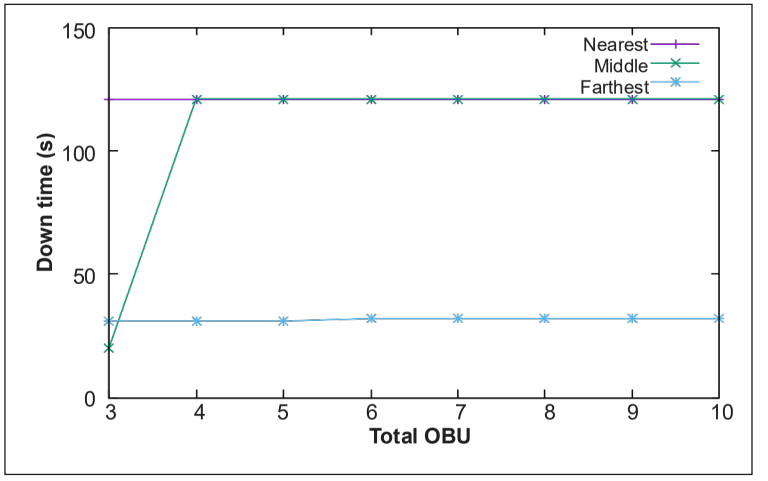


Figure 7. Failback downtime of link A down, using STP

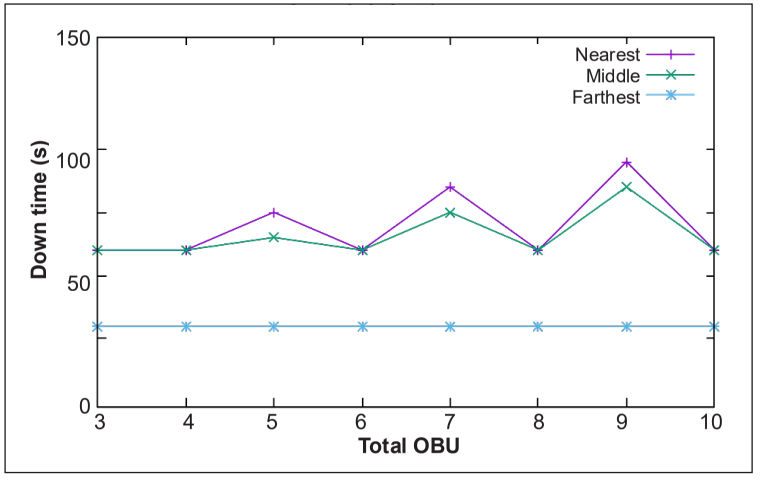


Figure 8. Failover downtime of link A down, using RSTP

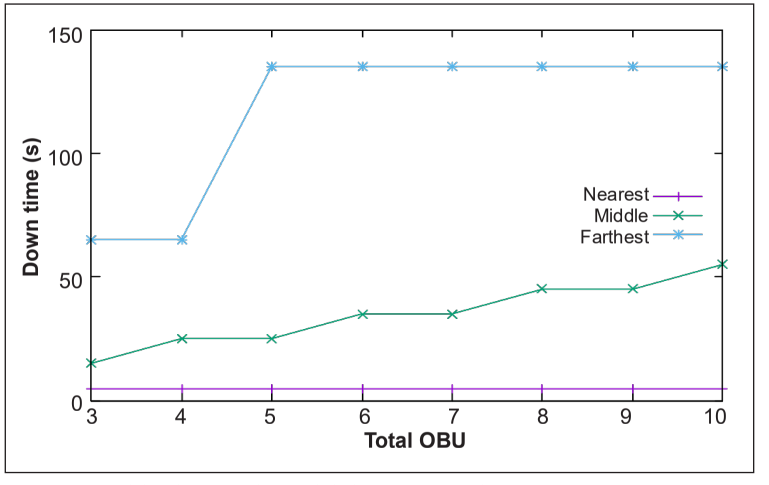


Figure 9. Failback downtime of link A down, using RSTP

Link Down at B

The results in the link B down scenario are only the nearest category at STP and RSTP. In this scenario, the one affected is only the OBU with an AP on the link, and it does not impact the path of the root bridge. As a result, the failover downtime on STP is around the 90 s longer than on RSTP. The downtime values on RSTP fluctuate, and the ones with the odd number of OBUs are relatively longer than those of the even number of OBUs (Figure 10). In the failback event, the comparison between the nearest STP and RSTP is relatively static, with a 90 s gap between them (Figure 11).

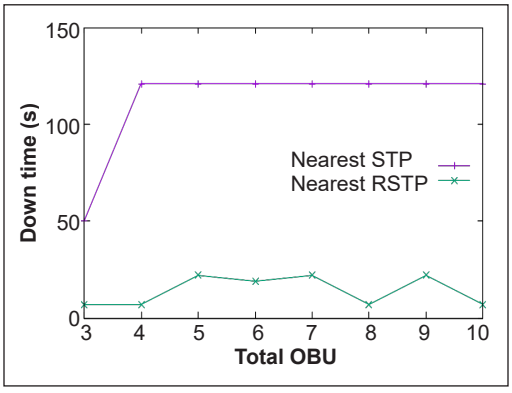


Figure 10. Failover downtime of link B down, comparing STP and RSTP

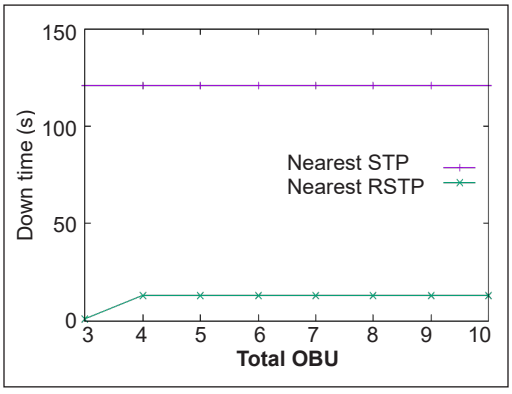


Figure 11. Failback downtime of link B down, comparing STP and RSTP

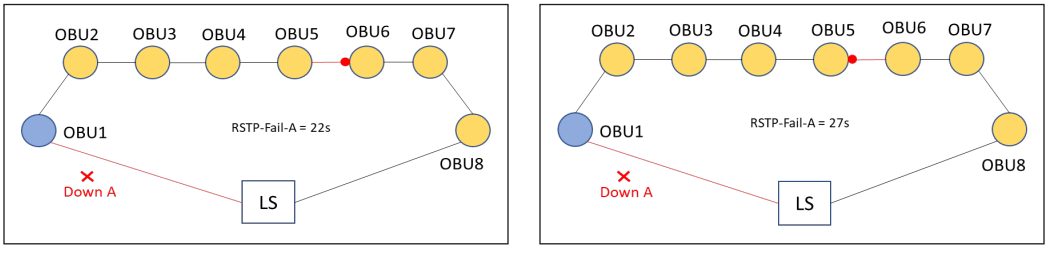
DISCUSSION

STP behavior patterns for both downtime scenarios are mostly the same. This condition is because STP uses a delay timer to determine changes to the state of the spanning tree rather than changing it immediately once the trigger comes. In RSTP, the blocked port status can change instantly when the topology change notification (TCN) is obtained from the incident point so that data traffic can be streamed immediately. So, RSTP's downtime is always lower than that of STPs.

RSTP has an interesting and different downtime pattern from STP, where its values fluctuate with the number of nodes, especially in failover scenarios due to the difference in distance from the incident point to nodes that have blocked ports in topologies that have an even or odd number of OBUs. When an incident occurs, the affected node will convey TCN to all nodes until it reaches the node with a blocked port and turns it into a forward state.

As shown in Figure 8, there is an interesting pattern where RSTP gives different downtime on odd and even OBUs. On odd OBUs, it shows higher downtime rather than that of even OBUs. It should be noted that the spanning tree configures all nodes inside the tree, including LS. The higher downtime in odd OBUs' situations is due to the farther distance needed to send a TC message from the failure point to the blocked port.

The reason above has been tested by taking the different positions of the blocked port in even OBUs [eight OBUs (Figure 12)]. In the default condition, the spanning tree will calculate bridge IDs to place the blocked port on OBU6 facing OBU5 since OBU5 has a lower bridge ID (by MAC address). Then, the downtime that occurs during link A failover is 22 s. When manually changing the bridge priority value on OBU5 to be higher than the default value (32768), the blocked port changes to OBU5 facing OBU6, and the downtime that occurs with the same scenario is 27 s.



(a) (b)

Figure 12. (a) Topology while OBU6 has a lower bridge ID as default; and (b) Topology while OBU5 has a lower bridge ID as manually configured

CONCLUSION

This paper has presented a further assessment of using STP and RSTP for INA-CBT OBU to LS communication segment using simulation. The proposed model is scaled up successfully until ten OBUs with a ring topology. The experimental results show that the convergence time values in failover and failback scenarios are still below the system requirement of INA-CBT. As expected, RSTP shows a faster convergence time than STP, but one should be aware of the possible RSTP downtime fluctuations depending on the number of OBUs deployed. Further investigation is needed to assess the system's dynamic behavior in scenarios with more complex topologies.

ACKNOWLEDGEMENTS

The authors thank the National Research and Innovation Agency (BRIN), Indonesia, for supporting this research project.

REFERENCES

Babu, S., & Kumar, P. A. R. (2022). A comprehensive survey on simulators, emulators, and testbeds for VANETs. *International Journal of Communication Systems*, *35*(8), Article e5123. https://doi.org/https://doi.org/10.1002/dac.5123

Domenikiotis, C., Loukas, A., & Dalezios, N. R. (2003). The use of NOAA/AVHRR satellite data for monitoring and assessment of forest fires and floods. *Natural Hazards and Earth System Sciences*, *3*(1/2), 115–128. https://doi.org/10.5194/nhess-3-115-2003

Hamdani, M., Irawan, D., Kusuma, A. A. N. A., Agastani, T., & Iqbal, M. (2023). Preliminary assessment of using spanning tree open protocols in INA-CBT communication system. In P. H. Khotimah (Ed.), *The 2022 International Conference on Computer, Control, Informatics and its Applications* (pp. 6–10). Association for Computing Machinery (ACM). https://doi.org/10.1145/3575882.3575884

Iqbal, M., Suwandi, B., Diana, M., Dewi, M. F., Herminawan, F. W., Giyana, R. F., Anggraeni, S. P., Firdaus, M. Y., Wibawa, Y. P., Palokoto, T. B., Utama, R. P., Marianto, F. A., Hamidah, M., Rahardjo, S., Purnomo, E., & Yogantara, W. W. (2021, November 8-9). *Performance analysis of Indonesia cable based tsunameter (INA-CBT) rokatenda ring topology*. [Paper presentation]. IEEE Ocean Engineering Technology and Innovation Conference: Ocean Observation, Technology and Innovation in Support of Ocean Decade of Science (OETIC), Jakarta, Indonesia. https://doi.org/10.1109/OETIC53770.2021.9733745

Iwata, A., Hidaka, Y., Umayabashi, M., Enomoto, N., Arutaki, A., Takagi, K., Cavendish, D., & Izmailov, R. (2004). Global open Ethernet architecture for a cost-effective scalable VPN solution. *IEICE Transactions on Communications*, *87*(1), 142–151.

Johnson, D. B., & Maltz, D. A. (1996). Dynamic source routing in ad hoc wireless networks. In T. Imielinski, & H. F. Korth (Eds.), *Mobile Computing* (pp.153–181). Springer https://doi.org/10.1007/978-0-585-29603-6_5

Kaiser, F., & Witthaut, D. (2021). Topological theory of resilience and failure spreading in flow networks. *Physical Review Research*, *3*(2), Article 23161. https://doi.org/10.1103/PhysRevResearch.3.023161

Kolarov, A., Sengupta, B., & Iwata, A. (2004, November 29 – December 3). *Design of multiple reverse spanning trees in next generation of Ethernet-VPNs*. [Paper presentation]. IEEE Global Telecommunications Conference, Dallas, USA. https://doi.org/10.1109/GLOCOM.2004.1378212

Kusuma, A. A. N. A., Agastani, T., Nugroho, T., Anggraeni, S. P., & Hartawan, A. R. (2022, December 6-7). *Estimating MQTT performance in a virtual testbed of INA-CBT communication sub-system*. [Paper presentation]. International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET), Bandung, Indonesia. https://doi.org/10.1109/ICRAMET56917.2022.9991213

Marchese, M., & Mongelli, M. (2012). Simple protocol enhancements of rapid spanning tree protocol over ring topologies. *Computer Networks*, *56*(4), 1131–1151. https://doi.org/10.1016/j.comnet.2011.10.008

Mikada, H., Hirata, K., Matsumoto, H., Kawaguchi, K., Watanabe, T., Otsuka, R., & Morita, S. (2003, June 25-27). *Scientific results from underwater earthquake monitoring using cabled observatories*. [Paper presentation]. International Conference Physics and Control. Proceedings (Cat. No.03EX708), Tokyo, Japan. https://doi.org/10.1109/SSC.2003.1224100

Mupparapu, S. S., Bartos, R., & Haag, M. (2005, August 21-24). *Performance evaluation of ad hoc protocols for underwater networks*. [Paper presentation]. Fourteenth International Symposium on Unmanned Untethered Submersible Technology (UUST'05), Durham, USA.

Perlman, R. (1985). An algorithm for distributed computation of a spanningtree in an extended LAN. *ACM SIGCOMM Computer Communication Review, 15*(4), 44–53. https://doi.org/10.1145/318951.319004

Perlman, R. (2000). *Interconnections: Bridges, Routers, Switches, and Internetworking Protocols*. Addison Wesley.

Premod, D. M., Kumar, K. S. A., Joseph, K. S., Kumar, B. K. P., Miny, G., & Shenoi, V. S. (2013, October 23-25). *Network design for submarine sonar systems*. [Paper presentation]. Ocean Electronics (SYMPOL), Kochi, India. https://doi.org/10.1109/SYMPOL.2013.6701937

Privadi, A., Damara, D. R., Widati, P. L., & Triputra, F. R. (2021, November 8-9). *Indonesia's cable based tsunameter (CBT) system as an earthquake disaster mitigation system in East Nusa Tenggara*. [Paper presentation]. IEEE Ocean Engineering Technology and Innovation Conference: Ocean Observation, Technology and Innovation in Support of Ocean Decade of Science (OETIC), Jakarta, Indonesia. https://doi.org/10.1109/OETIC53770.2021.9733734

Prytz, G. (2007, September 25-28). *Network recovery time measurements of RSTP in an ethernet ring topology*. [Paper presentation]. IEEE Conference on Emerging Technologies and Factory Automation (EFTA), Patras, Greece. https://doi.org/10.1109/EFTA.2007.4416924

Purwoadi, M. A., Anantasena, Y., Pandoe, W. W., Widodo, J., & Sakya, A. E. (2023, March 6-9). *Introduction to Indonesian cable-based subsea tsunameter*. [Paper presentation]. IEEE Underwater Technology (UT), Tokyo, Japan. https://doi.org/10.1109/UT49729.2023.10103368

Steinbach, T., Kenfack, H. D., Korf, F., & Schmidt, T. C. (2011, March 21-25*). An extension of the OMNeT++ INET framework for simulating real-time ethernet with high accuracy*. [Paper presentation]. Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques, Barcelona, Spain.

Varga, A. (2010). OMNeT++. In K. Wehrle, M. Gunes & J. Gross (Eds.) *Modeling and Tools for Network Simulation* (pp. 35–59). Springer. https://doi.org/10.1007/978-3-642-12331-3_3

Wang, T., Wang, X., Wang, Z., Guo, C., Moran, B., & Zukerman, M. (2021). Optimal tree topology for a submarine cable network with constrained internodal latency. *Journal of Lightwave Technology*, *39*(9), 2673–2683. https://doi.org/10.1109/JLT.2021.3057171

Widiyantoro, S., Gunawan, E., Muhari, A., Rawlinson, N., Mori, J., Hanifa, N. R., Susilo, S., Supendi, P., Shiddiqi, H. A., Nugraha, A. D., & Putra, H. E. (2020). Implications for megathrust earthquakes and tsunamis from seismic gaps south of Java Indonesia. *Scientific Reports*, *10*(1), Article 15274. https://doi.org/10.1038/s41598-020-72142-z

Zhang, L., Kim, T. H., Liu, C., Sun, M. T., & Lim, A. (2008, December 15-18). *Traffic-aware routing tree for underwater 3-D geographic routing*. [Paper presentation]. International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Sydney, Australia. https://doi.org/10.1109/ISSNIP.2008.4762036