International Journal Of Advanced Communication Technology (IJACT)
Volume 4, 2024 [47-61]

Performance Evaluation of Fault Tolerance in 6G Software Defined Network (SDN)

Lim Jia Chyin[1,]

[1]Faculty of Electronic Engineering & Technology, University Malaysia Perlis (UniMAP), 02600 Arau, Perlis, Malaysia

ABSTRACT

Traditional Operations and Maintenance (O&M) have made real-time fault location of devices challenging due to the rapid growth of network devices. Therefore, Intelligent O&M in 6G networks leveraged using Software Defined Network (SDN) are expected to optimise network operations through fault tolerance mechanisms. SDN is an emerging paradigm that offers dynamic and efficient solutions to the complex network environment. This paper provides a comprehensive overview of fault tolerance as a critical algorithm for SDN under Intelligent O&M, evaluating the performance of fault tolerance mechanisms and investigating the effectiveness of fault tolerance mechanisms towards TCP, UDP and ICMPv4 with two different scenarios. Simulation results demonstrate that the proposed fault tolerance mechanisms under Intelligent O&M within a 6G network can reduce the delay by 98.79%, ensuring more reliable network infrastructure.

Keywords: Operations and Maintenance (O&M), Fault Tolerance Mechanisms, Software-Defined Network (SDN), 6G.

1. INTRODUCTION

Next-generation 6G wireless communication networks are anticipated to provide smart devices and the Internet of Things (IoT) with higher speeds, lower latency and higher reliability to accommodate the explosive growth of end users across various applications. The intelligent information society will be characterised by highly digitised, intelligent and globally data-driven, enabled by near-instant and unlimited full wireless connectivity by 2030[1]. Intelligent O&M in 6G networks, together with Software Defined Network (SDN), is vital in accommodating the rise in network infrastructure by offering an innovative solution to various communication network challenges, such as workload increase associated with real-time monitoring.

Higher workloads often lead to challenges in efficiently managing and maintaining the network infrastructure, which may negatively impact overall performance and operational efficiency. Fault tolerance becomes a critical issue in 6G Intelligent O&M, especially considering diverse application requirements for reliability, latency and throughput. A failure occurs when the network cannot deliver a service correctly, while a fault is the root cause of a failure [12].

Traditional O&M often struggles with slow fault detection and location speed during network congestion, further contributing to the negative impact on communication networks. The manual process of identifying and resolving faults makes it difficult for network administrators to address issues effectively, leading to network performance degradation in terms of throughput, packet loss, delay and round-trip time.

Software Defined Networks (SDN) emerged as a new networking approach to solve the issues. SDN introduces a software-driven approach using software-driven controllers and application programming interfaces (APIs) to manage underlying hardware infrastructure and direct traffic

*jiachyin0701@gmail.com

across the network. SDN architecture separates the control plane- which makes decisions about traffic routing, and the data plane – which handles packet forwarding between network devices and emerged as one of the most important network paradigms in recent years [2].

SDN offers a more flexible, automated and efficient network management by decoupling the control and data planes. Integrating Intelligent O&M in SDN within 6G networks provides a more adaptive solution to address traditional O&M challenges by ensuring the network remains available and responsive to user demands despite faults.

In early 2016, Gonzalez, Nencioni, Helvik and Kaminski [3] discussed basic SDN architecture or single-point SDN controller, depicting the control plane as a potential single point of failure where the controller was not fast enough for the capability of a network to handle new flow is limited considerably affect users and provide a roadmap for their research in designing and developing Master-Slave SDN controller in handling the case of failures. Master-slave SDN controller is a simpler concept in charge of all decisions, while the backup controller provides fault tolerance when the master breaks down.

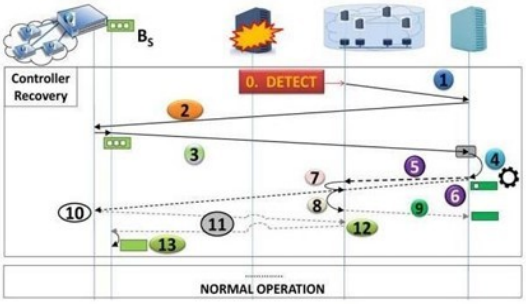

Figure 1. Sequence Diagram Master Failure Recovery Routine [3]

Botelho, Bessani, Ramos and Ferreira [4] have presented that the increase in SDN-based deployments in production networks is triggering the need to consider fault-tolerant design of controller architectures. They contributed to designing fault tolerance and consistent SDN controllers known as SmartLight Architecture. It uses the Paxos Algorithm to implement the data store as a Replicated State Machine (RSM).

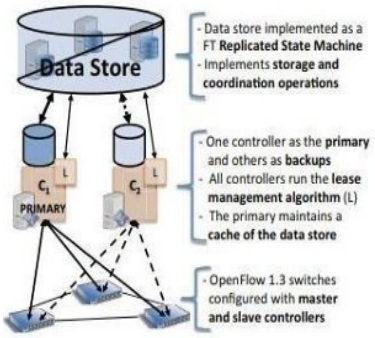

Figure 2. SmartLight Architecture [4]

Aly and Kotb [10] published research titled "Towards SDN Fault Tolerance using Petri-Nets," which presented new core features in SDN, including centralised control and programmability. In this research, the Fault Tolerance using the Petri-Nets for SDN networks (FTPNSDN) model has been proposed to increase the network's availability compared to a reference model and show stability after failures. The results proved that packet delays for this model have greatly reduced by 12% compared to the Hyper Flow reference model.

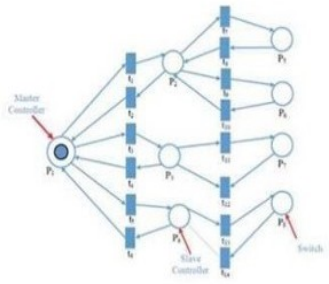

Figure 3. Petri-Nets Structure SDN [10]

Kim, Santos, Turner, Schlansker, Tourrilhes, and Feamster [13] have greatly presented that the existing solutions from legacy networks do not work out-of-the-box in SDN networks, and pure SDN-based solutions are not efficient. In their research, CORONET, an SDN fault-tolerant system, has been proposed to recover from multiple link failures in the data plane. At the same time, this research describes a prototype implementation based on NOX, which demonstrates fault recovery for emulated topologies using Mininet.

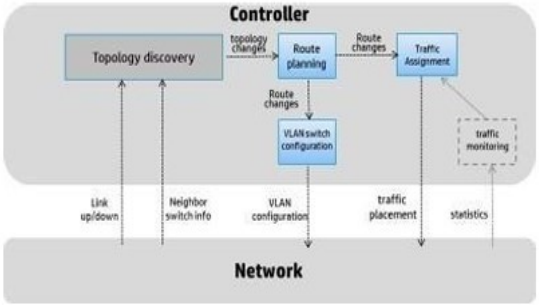

Figure 4. CORONET Architecture [13]

Yamansavascilar, Baktir, Ozgovde and Ersoy [15] in the journal have presented fault tolerance in the SDN Data Plane where network failures caused service disruptions. This situation deteriorates the Quality of Service (QOS) for the SDN to become the mainstream network paradigm. In this research, BFD with Dynamic Protection with Quality Alternative Paths (DPQoAP) has been proposed to enhance the adaptability and efficiency of the networking operations of the applications.

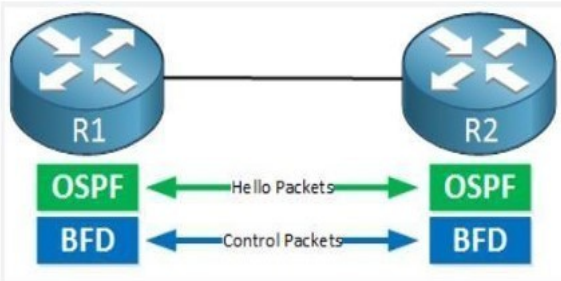

Figure 5. Bidirectional Forwarding Detection BFD Protocol [15]

Barakabitze and Walshe [16] in the journal have greatly accomplished and made a huge contribution to how future communication 6G systems will meet network performance targets such as lower cost, lower latency, higher capacity and satisfied Quality of Service (QoS). This

journal also provides a comprehensive and ground-based approach to Quality of Experience (QoE) management in the context of future software 6G networks.

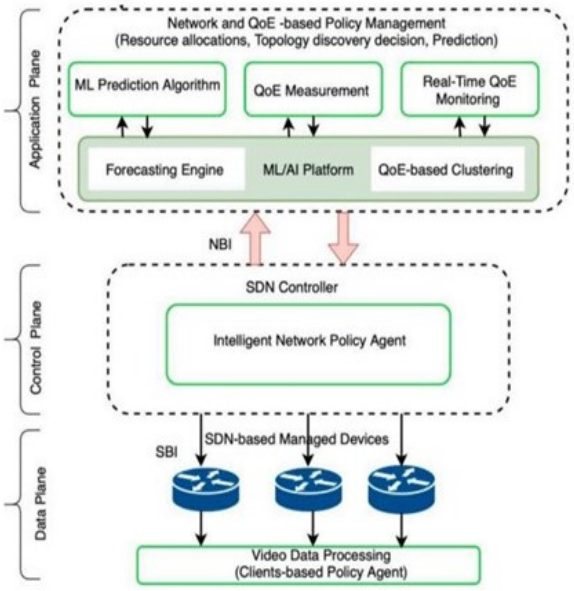

Figure 6. 6G SDN controller for quality of service Provisioning [16]

2. TERMINOLOGY

2.1 Sixth-Generation (6G)

Sixth-Generation (6G) extensively uses edge platforms that integrate communications, control and processing resources, offering high speed and low latency. It's a technology that leads in front of Generation (5G). It makes 6G more suitable for applications such as autonomous vehicles, augmented reality (AR), autonomous healthcare solutions and others. Holographic telepresence, eHealth including in-body networks, is another few 6G use cases [5] that demand extremely high data rates, ultra-low latency and ultra-reliability. The evolution of 6G application domains requires an innovative architecture beyond current network designs [6].

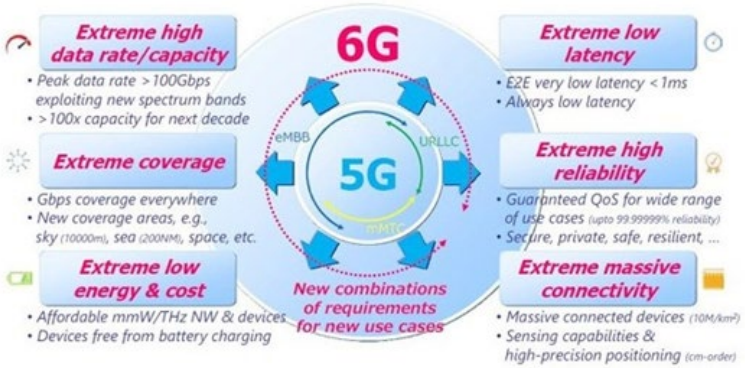

Figure 7. Requirements of Sixth-Generation Technology

2.2 Software Defined Network

Software-defined network (SDN) is a networking paradigm separating the control and data planes in traditional networks. The control plane is the layer responsible for deciding where and how data packets should be routed through the network. In contrast, the data plane is the layer where the actual forwarding of data packets is carried out based on decisions made by the control plane. SDN is a centralised and single controller point in managing the network devices. As Douha [7] mentioned, SDN enhances the management and access control of the home network by providing a programmable controller to home nodes. SDN has been widely integrated into various networking environments such as data centres, enterprise networks and wide-area networks.

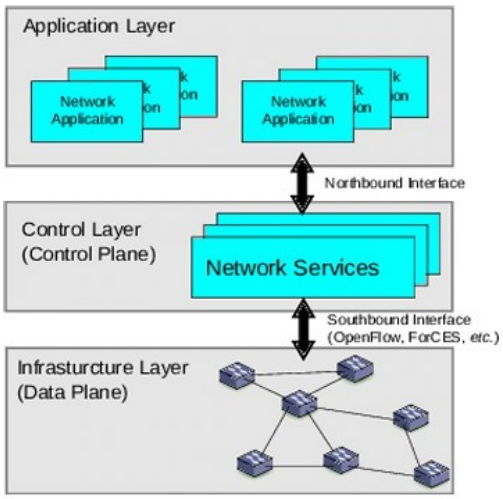

Figure 8. Software Defined Network Architecture [11]

2.3 Fault Tolerance

Fault tolerance is the ability of a system to continue working properly with minimal disruptions despite there is presence of failures. Fault tolerance is vital in ensuring the consistency, reliability and availability of the network, as there is a possibility that network devices and communication links can be prone to various types of failures, such as hardware failures, software failures or link failures. Redundancy is the main approach to duplicating network devices. The redundant device can take over the operations seamlessly if any components fail. Additionally, resiliency is another approach to dynamically adapting and recovering from failures, such as by rerouting traffic or reconfiguring network resources. Gonzalez, Nencioni, Helvik, and Kamisin [3] recently published research titled Fault Tolerant Master-Salve SDN Controllers where Distributed Datastore (SMR) method is applied, the primary devices will frequently synchronise the data for the whole topology in case there is a new device coming in.

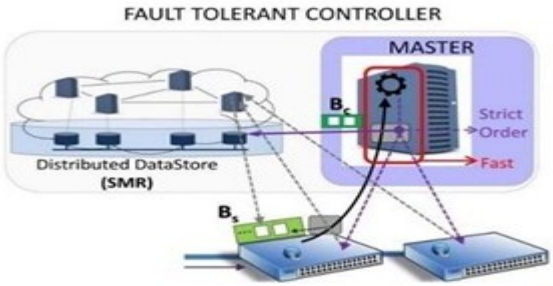

Figure 9. Fault Tolerance Master-Slave SDN Controller [3]

2.4 6G Intelligent Operations & Maintenance (O&M)

The emergence of 6G Intelligent Operations and Maintenance (O&M) introduces a paradigm shift in the management and optimisation of next-generation communication networks. 6G includes traditional on-ground vehicular-to-vehicular/infrastructure communications and air-ground, space-terrestrial and even underwater vehicles. [8] There are a few examples of 6G Intelligent, which are Intelligent Manufacturing and Intelligent O&M. Intelligent manufacturing based on digital twins can independently collect and analyse production data and carry out an inference, prediction, independent planning, and decision-making production process via approaches like digital imitation and comprehensive simulation.[9] It emphasises its role in detecting, locating, and rectifying faults based on knowledge graphs and expert experience.

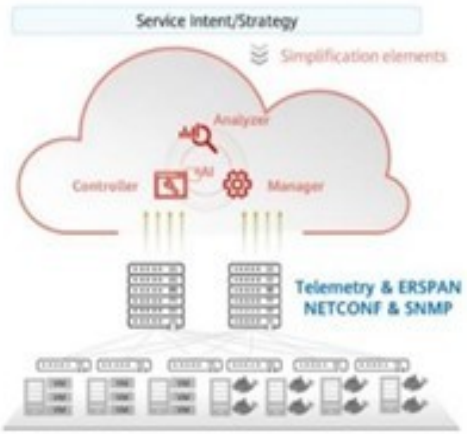

Figure 10. 6G Intelligent O&M SDN Architecture

3. METHODOLOGY

3.1 Simulation Tools

Simulation tools are discussed to develop network topology and evaluate fault tolerance performance in a 6G Software Defined Network (SDN). Various open-source simulation tools are available, such as Mininet, OMNeT++, Ns-2/ Ns-3 and GNS 3. It allows users to simulate and evaluate fault tolerance mechanisms through redundancy or backup SDN controllers. Mininet creates a virtual network that will enable users to test how different network devices behave without real devices; it essentially provides an infrastructure layer in the network. On the other hand, Open Daylight is an open-source SDN controller in the control layer that decides how traffic should be routed. Mininet is integrated with Open Daylight to evaluate fault tolerance performance in 6G SDN.

Figure 11. Mininet and Open Daylight

3.2 Data Collection Techniques (Performance Metrics)

Performance metrics are important in objectively measuring the network's performance and assessing its behaviour under various conditions. Various performance metrics (packet loss ratio, throughput, delay and round-trip time) are used to measure the effectiveness and efficiency of the proposed fault tolerance mechanisms in 6G SDN in maintaining minimal packet delivery disruption even with the presence of faults.

3.2.1 Packet Loss Ratio

A lower packet loss ratio indicates a better performance as it shows a higher percentage of packets reaching their desired destinations without being dropped or lost due to faults in the network. The formula packet loss ratio can be written as:

$$Packet\ Loss\ Ratio = 100\% - \frac{100\ x\ (\ Total\ Packets - Dropped\ Packets\ )}{Total\ Packets} \quad (1)$$

3.2.2 Throughput

The measure of the amount of data successfully transmitted over the network per unit of time is known as throughput. It indicates the capacity and efficiency of the network in handling traffic. Higher throughput indicates better performance for this network, reflecting its ability to transmit 42 data without congestion or bottleneck issues efficiently. Throughput is usually measured in bits per second (bps) or packets per second (pps), and the formula can be written as:

$$Throughput = \frac{Total\ Amount\ Data\ Transmitted}{Time\ Taken\ for\ Transmission} \quad (2)$$

3.2.3 Delay

Delay measures the time a packet takes to travel from the source to the destination in a network. It describes the latency and responsiveness of the network. Lower delay indicates better performance, which means faster packet transmission and lower waiting times. Delay is measured in several methods, which are one-way delay, round-trip delay or average delay, and the formula can be written as:

$$One - Way\ Delay = Time\ taken\ for\ a\ packet\ to\ travel\ from\ source\ to\ dest. \quad (3)$$

$$Round - Trip\ Delay = Time\ taken\ for\ a\ packet\ to\ travel\ two\ ways\ source\ to\ dest. \quad (4)$$

$$Average\ Delay = Average\ tim\ taken\ for\ packet\ to\ travel\ from\ source\ to\ dest. \quad (5)$$

3.3 Data Flowchart for Fault Tolerance in 6G SDN

The flowchart will further discuss how the fault tolerance in 6G SDN works. First, the SDN controller will retrieve basic network topology information and monitor the network congestion condition/link failure. SDN controller will determine the source of congestion and the link failure using Algorithm 1 (Fault Tolerance Algorithm), which will be elaborated after. If there is no congestion/ link failure after the execution of Algorithm 1, the SDN controller will continue to monitor to determine network congestion/link failure condition. Suppose the detection shows that the link in the network is facing congestion. In that case, it will proceed to the method of mitigating the congestion using the fault tolerance mechanism, which is load balancing the flow across the multiple paths using Equal Cost Multi-Path (ECMP) where there is a Dijkstra algorithm

helping in selecting the shortest path between the last of the congested node towards the destination node. After the algorithm is executed, the network congestion condition is solved.

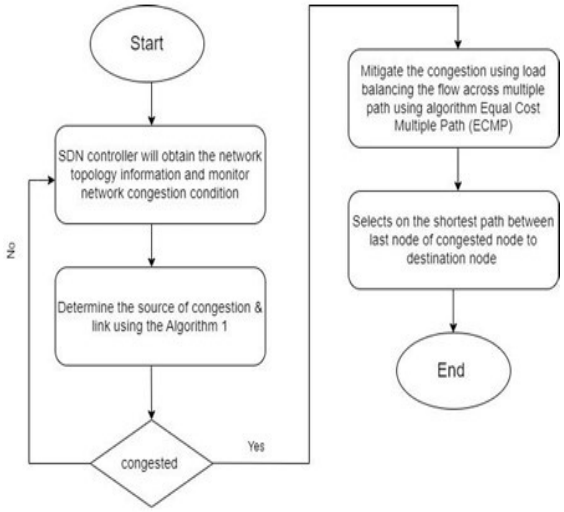

Figure 12. Flowchart for Fault Tolerance in 6G SDN (Network Failure)

3.4 Network Congestion Condition/ Link Failure Detection Algorithm

The network Congestion Condition/ Link Failure Detection Algorithm starts by initialising the value of M, which stands for the normal status at first. It runs the if-else statement where messages are injected with normal status and undefined-f ring direction that plays a vital role in guiding the packet to reroute to another alternative path either in the f-ring direction of clockwise or anti-clockwise If M normal hop is on the faulty link. M's status will need to be set the M'status as misrouted and the M-f-ring direction based on clockwise and counterclockwise being set in the Routing Table. Once the direction is found to reroute the packet, it makes sure that M'type is DIM1+ and M's normal hop is DIM1+ next hop; if not, it will proceed to another line where M'type DIM1- and normal hop is DIM1- hop and M's normal hop is not blocked then it automatically set the M'type to Normal which means that the packet is successfully rerouted to another alternative path with good link conditions and no link failures. The M's f-ring status becomes undefined as it does not need to guide the packet either routing clockwise or anti-clockwise. Then it will go back to the first cycle, where it will monitor the M's normal hop to see whether on the faulty link; if yes, then it continues to walkthrough again the process.

```
// Messages are injected with normal status and undefined f-ring direction
// When a node receives a message meant for it, the node consumes the message
// Otherwise, it applies the following algorithm to determine
// the next node in message's path
1.  IF (M's status is normal) // M is currently normal. See if it should continue to be normal
    THEN
        Determine the dimension of the normal hop for M. Let it be i.
        IF (M's normal hop is on a DIM_i+ channel) THEN
            set M's type to DIM_i+
        ELSE set M's type to DIM_i-
        ENDIF
        IF (M's normal hop is on a faulty link) THEN
            set M's status to misrouted
            set M's f-ring direction to clockwise or counter-clockwise using Table 1
        ENDIF
2.  ELSE // M is being misrouted. See if it should become normal
        IF ((M's type is DIM_i+ AND M's normal hop is a DIM_i+ hop)
          OR (M's type is DIM_i- AND M's normal hop is a DIM_i- hop)
          OR ((M's type is DIM_0+ OR DIM_0-) AND M's normal hop is not blocked))
        THEN
            set M's type to Normal
            set M's f-ring direction to undefined
        ENDIF
    ENDIF
// Route M using e-cube or misrouting logic
3.  IF (M's status is normal) THEN
        use the normal hop to route the message
    ELSE use the hop in the f-ring direction specified in its status field
    ENDIF
```

Figure 13. Algorithm 1- Network Congestion Condition / Link Failure Detection Algorithm

3.5 Topology Design

Topology design is a vital aspect that needs to be included in this research because it determines the overall network structure architecture. This is a multipath network topology with 5 hosts and 6 OpenFlow switches, where host 1 and host 2 have three path selections to forward the packets to host 3, host 4 and host 5, respectively. The Open Daylight SDN controller in the control layer communicates with the Open-Flow switches in the infrastructure layer through the Southbound Interface (SBI) with OpenFlow Protocol to give instructions on selecting the optimal and shortest paths.

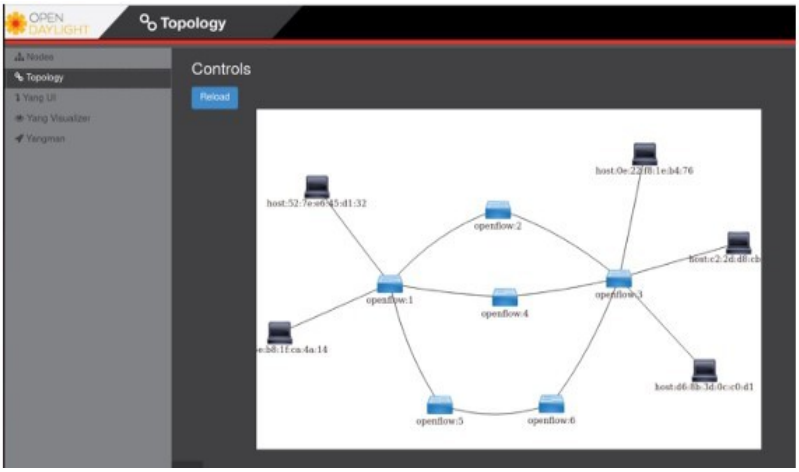

Figure 14. Multipath Network Topology in Open Daylight

3.6 Multipath Dijkstra with Node Count

3.6.1 Initialization

- Setting dist[v] to infinity (∞), visited[v] to FALSE, previous[v] to UNDEFINED, and node_count[v] to infinity for all vertices v in graph G.
- Setting dist[s] and node_count[s] to 0 for the source node s.

3.6.2 Inserting Source

- Inserting source nodes into the priority queue Q.

3.6.3 Processing Queue

- While the priority queue Q is not empty: label=0.
  a) Selecting Vertex:
     - Selecting the vertex u in Q with the smallest dist[u] that hasn't been visited yet.
     - Removing you from Q and marking it as visited.
  b) Calculating Alternative Path:
     - For each neighbour v of you:
     - Calculating the alternative distance alt =dist[u] + dist_between(u, v).
     - Incrementing the node count for v as node_count[u] + 1.

c) Updating Shortest Path:
   - If alt < dist[v] or (alt=dist[v]and nodes <node_count[v]):
     - Updating dist[v] to alt.
     - Updating node_count[v] to nodes.
     - Resetting previous[v] to you.
   - Else, if alt = dist[v] and nodes =node_count[v]:
     - Add u to previous[v].

d) Inserting Neighbour:
   - If v has not been visited, insert it into Q.

3.6.4 Returning Result

• Returning the dist[] array containing the shortest distances and the node_count[] array with the corresponding node counts from the source node s to all vertices v in graph G.

Algorithm 2: Multipath Dijkstra with Node Count

Input: Graph G, source node s
Result: Shortest distances $dist[v]$ and node counts $node\_count[v]$ from source s to all vertices v in G
for *each vertex v in Graph* **do**
  $dist[v] \leftarrow \infty$;
  $visited[v] \leftarrow$ FALSE;
  $previous[v] \leftarrow$ UNDEFINED;
  $node\_count[v] \leftarrow \infty$;
end
$dist[s] \leftarrow 0$;
$node\_count[s] \leftarrow 0$;
Insert s into priority queue Q;
while *Q is not empty* **do**
  $u \leftarrow$ vertex in Q with smallest $dist[u]$ and not yet visited;
  Remove u from Q;
  $visited[u] \leftarrow$ TRUE;
  for *each neighbor v of u* **do**
    $alt \leftarrow dist[u] + dist\_between(u, v)$;
    $nodes \leftarrow node\_count[u] + 1$;
    if $alt < dist[v]$ ***or*** $(alt == dist[v]$ ***and*** $nodes < node\_count[v])$ **then**
      $dist[v] \leftarrow alt$;
      $node\_count[v] \leftarrow nodes$;
      Reset $previous[v]$ to u;
    else
      if $alt == dist[v]$ ***and*** $nodes == node\_count[v]$ **then**
        Add u into $previous[v]$;
      end
    end
    if *$visited[v]$ is **FALSE*** **then**
      Insert v into Q;
    end
  end
end
return $dist$, $node\_count$;

Figure 15. Algorithm2- Multipath Dijkstra with Node Count

4. RESULTS AND DISCUSSION

The results of this research reflect decent improvements in network performance, particularly in network delay reduction when applying SDN in 6G Intelligent Operations and Maintenance. By implementing fault tolerance mechanisms in 6G Intelligent O&M with SDN, the 6G network can automate and perform intelligent responses towards situations like link failures, network congestion and device malfunctions. These algorithms enable the system to have rapid detection and rectification based on expert experiences and simulation analysis. Utilising Algorithm 2- Multipath Dijkstra with Node Count where it identifies new path within seconds to ensure minimal disruptions to network operations during fault scenarios. The simulation framework

developed for this research reflects network resilience and efficiency in maintaining optimal performance despite adverse conditions.

4.1 Preliminary Results for 6G SDN and Traditional Network

Bandwidth (Mbits/sec)	Delay-ms (SDN)	Delay-ms (Traditional)
100	0.1821	1.1828
200	0.0929	0.3435
300	0.0968	1.2831
400	0.4427	1.1800
500	0.0237	1.9550

Table 1 Delay for 6G SDN and Traditional O&M

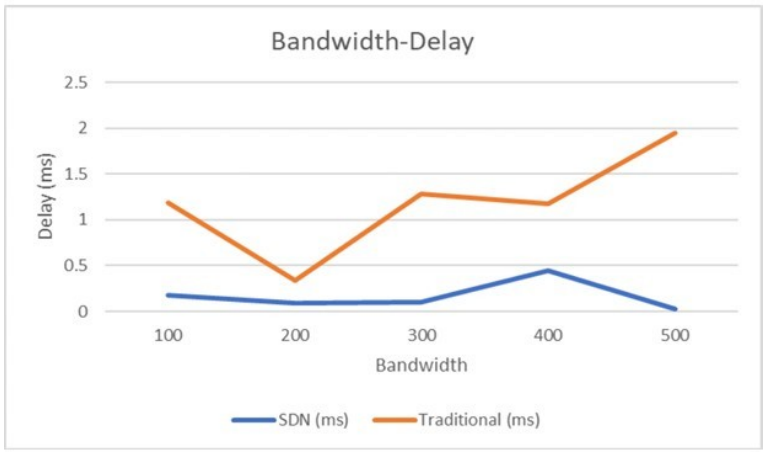

Figure 16. Bandwidth Delay for 6G SDN and Traditional O&M

As illustrated in Table 1 and Figure 16, the delay times in 6G Intelligent O&M with SDN are subsequently lower compared to Traditional O&M based on bandwidth. For example, at a bandwidth of 100 Mbits/sec, the delay for a 6G SDN system is 0.1821ms, while Traditional O&M experiences a higher delay of 1.1828ms. This shows a significant 73.32% delay when applying SDN. This improvement is primarily attributed to the SDN's superior fault detection and automatic rerouting capabilities, which prompt responses to link failures compared to the traditional O&M approach.

4.2 Discussion

Integrating two algorithms - Network Congestion Condition/ Link Failure Detection Algorithm Fault Detection and Multipath Dijkstra with Node Count- has contributed to substantial improvements, particularly in fault tolerance, rerouting efficiency and traffic delivery. Fault Detection and Rerouting Algorithms consistently play a vital role in monitoring the network in real-time for congestion and link failures, allowing it to dynamically reroute packets when failed

links or congestion-prone areas occur. These proactive rerouting measures minimise delays and ensure efficient traffic delivery where the network can bypass trouble spots and continue operation without disruptions, contributing to higher network reliability. Multipath Dijkstra with Node Count algorithm is designed to calculate the shortest and most efficient paths for data travel within the network. The algorithm allows the network to switch to an alternative path with multiple potential routes. The priority queue mechanism updates the path in real-time by reducing the search time for optimal routes. The combination of these algorithms strengthens the network's ability to respond to faults faster and maintain a higher-performance network through coupling proactive fault detection (link failures and congestion detection).

Although integrating these algorithms greatly enhances network performance, scalability and size pose potential pitfalls and challenges. The computational burden in maintaining multiple paths, rerouting traffic and continuously monitoring link failures could increase exponentially as the network scales to accommodate the much larger and more complex topologies of real-world 5G/6G systems. There might be a performance bottleneck due to the overhead of constantly recalculating the paths and maintaining multiple routes across thousands of nodes. Optimising algorithms for scalability is a great solution for addressing the scalability and network size issues through distributed computing techniques or parallel processing. These techniques help to divide the workload of path calculations and fault detections across multiple servers and nodes. The system can handle larger networks with minimal packet loss in network performance.

5. CONCLUSION

This research provides critical insights into the performance evaluation of fault tolerance in 6G Intelligent O&M with SDN. The findings successfully demonstrate that 6G SDN performs better than traditional O&M in terms of network delay with a great reduction of 98.79% from 1.9550ms to 0.0237ms at a bandwidth of 500 Mbits/sec. These substantial delay reductions align with the study objectives of this study, which include the successful identification of fault tolerance mechanisms, the ability to reroute traffic accordingly during link failures and the integration of these mechanisms with SDN within a 6G environment.

Implementing an SDN-based fault tolerance mechanism in a 6G network has significantly improved resilience and performance compared to traditional O&M. By incorporating an Open Daylight SDN controller and Mininet, automated packet rerouting when link failures occur ensures uninterrupted network operation despite faults. These findings underscore the potential of SDN to enhance network robustness, minimise downtime and reduce service interruptions in improving network performance and user experience.

6G Intelligent O&M with SDN framework offers faster network recovery, improving system reliability by rapidly detecting and rectifying faults. This capability is critical for high-demand applications that require continuous connectivity, such as autonomous vehicles, remote healthcare and IoT systems. The research supports a higher demand for intelligent fault tolerance in next-generation networks by showing that SDN-based mechanisms can improve network performance.

This research has demonstrated the effectiveness of fault tolerance in SDN within 6G networks; future research is needed to delve further deeper into other areas of network management, particularly in Policy Based Routing (PBR) in Software Defined Networks (SDN). It holds great potential to enhance the security and efficiency of SDN-based networks, while PBR is beyond the scope of this study. PBR involves several processes, which include routing decisions for the next hop process and policies defined by network administrators. PBR could provide fine-grained control over traffic flows, ensuring higher priority and sensitive data are routed first according to network performance requirements.

Future research circulates the integration of PBR with SDN within 6G networks and focuses on the development of dynamic and policy-driven routing algorithms. An additional PBR algorithm could address challenges despite network congestion, security and data allocation while also ensuring dynamic adaptive routing for various network conditions. Prioritising data security and network resilience with PBR is essential in protecting proprietary and integrity information in enterprise networks as SDN within 6G networks grows.

Fault tolerance mechanisms explored in the research have a wider-reaching application in the real-world deployment of 6G networks. For example, industries like autonomous driving, smart cities and remote healthcare could greatly enhance SDN's ability to ensure non-disruption connectivity and resilience against network failures. The SDN-based performance promises significant advantages; its real-world application faces challenges like the need for scalability in network environments.

Furthermore, the dynamic nature of 6G networks with higher traffic volume and lower latency requirements presents a new challenge in maintaining a robust network by ensuring seamless and efficient traffic rerouting during failures, especially in critical infrastructure, which requires further refinement of SDN protocols and algorithms. While the study highlights the greatest potential in improving fault tolerance in 6G networks, ongoing research is adequate to address the complexities of real-world network management and optimise these mechanisms for practical applications in diverse applications in diverse industries.

REFERENCES

[1] M. Latva-aho, "Radio access networking challenges towards 2030," ITU, 2018. [Online]. Available: https://www.itu.int/en/ITU-T/WorkshopsandSeminars/201810/Documents/MattLatvaahoPresentation.pdf.

[2] N. Feamster, J. Rexford, and E. Zegura, "The Road to SDN," Queue, vol. 11, no. 12, 2013, doi: 10.1145/2559899.2560327.

[3] A. J. Gonzalez, G. Nencioni, B. E. Helvik, and A. Kamisin´ski, "A fault-tolerant and consistent SDN controller," in Proc. IEEE Global Communications Conf. (GLOBECOM), 2016, pp. 1-6, doi: 10.1109/GLO- COM.2016.7841496.

[4] F. Botelho, A. Bessani, F. M. V. Ramos, and P. Ferreira, "On the design of practical fault-tolerant SDN controllers," in Proc. 3rd Eur. Workshop Software Defined Netw. (EWSDN), 2014, pp. 73-78, doi: 10.1109/EWSDN.2014.25.

[5] C. De Alwis et al., "Survey on 6G Frontiers: Trends, Applications, Requirements, Technologies and Future Research," IEEE Open Journal of the Communications Society, vol. 2, pp. 836-886, 2021, doi: 10.1109/OJ- COMS.2021.3071496.

[6] M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, "Toward 6G Networks: Use Cases and Technologies," IEEE Communications Magazine, vol. 58, no. 3, pp. 55-61, 2020, doi: 10.1109/MCOM.001.1900411.

[7] N. Y. R. Douha et al., "A survey on blockchain, SDN and NFV for the smart-home security," Internet of Things, vol. 20, 2022, doi: 10.1016/j.iot.2022.100588.

[8] S. Saharan, S. Bawa, and N. Kumar, ”Dynamic pricing techniques for Intelligent Transportation Systems in smart cities: A systematic view,” Computer Communications, vol. 150, pp. 603-625, 2020, doi: 10.1016/j.comcom.2019.12.003.

[9] X. Huang, ”Intelligent remote monitoring and manufacturing system of production line based on industrial Internet of Things,” Computer Communications, vol. 150, pp. 421-428, 2020, doi: 10.1016/j.comcom.2019.12.011.

[10] W. H. F. Aly and Y. Kotb, ”Towards SDN Fault Tolerance using Petri Nets,” in Proc. 2018 28th International Telecommunication Networks and Applications Conference (ITNAC), Sydney, NSW, Australia, 2018, pp. 1-6. doi: 10.1109/ATNAC.2018.8615188.

[11] W. Braun and M. Menth, ”Software-Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choices,” Future Internet, vol. 6, no. 2, pp. 302-336, 2014. doi: 10.3390/fi6020302.

[12] H. Zeng, P. Kazemian, G. Varghese, and N. McKeown, ”Automatic test packet generation,” in Proc. 8th Int. Conf. Emerging Networking Experiments and Technologies (CoNEXT ’12), Nice, France, 2012, pp. 241-252.

[13] H. Kim, J. R. Santos, Y. Turner, M. Schlansker, J. Tourrilhes, and N. Feamster, ”CORONET: Fault Tolerance for Software Defined Networks,” presented at the IEEE International Conference on Network Protocols, Austin, TX, USA, 2012.

[14] L. Ochoa-Aday, C. Cervello´-Pastor, and A. Ferna´ndez-Ferna´ndez, ”Current Trends of Topology Discovery in OpenFlow-based Software Defined Net- works,” 5GCity Project, 2015. doi: 10.13140/RG.2.2.12222.89929.

[15] B. Yamansavascilar, A. C. Baktir, A. Ozgovde, and C. Ersoy, ”Fault Tolerance in SDN Data Plane Considering Network and Application Based Metrics,” Journal of Network and Computer Applications, vol. 157, 2019. doi: 10.1016/j.jnca.2020.102780.

[16] A. A. Barakabitze and R. Walshe, ”SDN and NFV for QoE-driven multimedia services delivery: The road towards 6G and beyond networks,” Computer Networks, vol. 214, 2022. doi: 10.1016/j.comnet.2022.109133.