Adaptive Dynamic Bandwidth Allocation in Smart Cities Using Software-Defined Networking

Renaldy Fredyan
Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia
renaldy.fredyan@binus.ac.id

Luthfan Hadi Pramono
Computer Engineering Department, Faculty of Information Technology, Universitas Teknologi Digital Indonesia, Yogyakarta, Indonesia
luthfanhp@utdi.ac.id (corresponding author)

Khairunnisa
Computer Science Department, Faculty of Engineering and Computer Science, Universitas Muhammadiyah Bima, Bima, Indonesia
khairunnisa@umbima.ac.id

Received: 15 February 2025 | Revised: 3 March 2025 and 20 March 2025 | Accepted: 23 March 2025

ABSTRACT

The rapid expansion of smart city applications has led to an increased demand for real-time data processing and optimal resource utilization. However, traditional static bandwidth allocation techniques often fail to meet the changing needs of these services. This research aims to provide an alternative solution through adaptive dynamic bandwidth allocation based on Software-Defined Networking (SDN) to enhance bandwidth distribution in smart cities. The proposed system utilizes POX software to dynamically allocate bandwidth based on real-time network conditions and service requirements, leveraging SDN's programmability and central control capabilities. The system prioritizes network allocation according to device priority and bandwidth needs to improve overall network efficiency. The findings of the simulation demonstrate that adaptive dynamic bandwidth allocation using SDN significantly improves network performance compared to conventional static techniques, reducing latency and enhancing Quality of Service (QoS) for critical smart city applications, such as traffic control and public safety. This approach presents a promising alternative for future smart city network infrastructure.

Keywords-bandwidth allocation; SDN; smart city network

I. INTRODUCTION

Smart cities address the challenges of urbanization by integrating technology to improve quality of life, enhance security, and optimize resource management [1]. Communication networks play a vital role in supporting applications such as Closed-Circuit Television (CCTV), sensors, traffic systems, and emergency services. However, traditional network management struggles with dynamic bandwidth demands, causing congestion during simultaneous use [2]. Software-Defined Networking (SDN) enables dynamic, adaptive bandwidth allocation to meet real-time demands, enhancing efficiency and preventing congestion. As smart cities grow, reliable real-time data processing and efficient utilization of resources are crucial to maintaining low operational costs. Increasing demand for services like public internet, emergency communication, and CCTV surveillance highlights the limitations of traditional static bandwidth allocation, which often leads to inefficient resource use and poor service quality [3]. SDN addresses these issues by separating control and data delivery functions, centralizing management, and enabling flexible resource allocation [4]. Virtualization further enhances scalability and adaptability by replacing dedicated network hardware with general-purpose devices, reducing costs and improving efficiency [5-7]. In video surveillance, a critical component of smart city infrastructure, SDN enables flexible, scalable systems. Authors in [8] demonstrate how SDN facilitates large-scale Internet Protocol (IP) camera deployment, dynamic video feed routing, and configurable services like motion detection and encryption.

Existing research has extensively explored the integration of SDN in enhancing smart city applications [9, 10]. Authors in [11] investigate middlebox virtualization, emphasizing programmability and scalability in modern surveillance. Authors in [12] propose adaptive video coding for wireless networks, introducing an intelligent algorithm that dynamically adjusts video quality based on available bandwidth, ensuring reliable video transmission in smart city surveillance [13]. Other studies [14, 15] examine SDN-based network virtualization, highlighting its benefits in terms of energy efficiency, cost reduction, and service agility. While these approaches focus on scalability, energy optimization, or video quality, they lack an emphasis on real-time bandwidth allocation tailored to dynamic service demands. The present study aims to address this gap by proposing a system that prioritizes bandwidth allocation based on service urgency, ensuring reduced latency, improved Quality of Service (QoS), and efficient network resource management for smart city applications [16-22].

The objective of this study is to propose a system that focuses on optimizing bandwidth allocation for smart city services by prioritizing three service categories: high priority for CCTV surveillance to maintain real-time video feeds, medium priority for emergency services requiring stable communication, and low priority for public internet access with flexible bandwidth allocation. By leveraging the POX SDN controller, the system dynamically adjusts bandwidth based on application urgency and relevance, ensuring optimal resource utilization. This approach enhances network efficiency, minimizes latency, and improves QoS. Designed to be scalable and adaptive, the system effectively addresses the increasing complexity of smart city infrastructures, providing real-time bandwidth management tailored to diverse service demands.

II. PROPOSED APPROACH

The proposed project involves the creation and implementation of an adaptive dynamic bandwidth allocation system based on SDN, with a particular emphasis on prioritizing key smart city services. Starting with the configuration of a simulation environment, the project approach consists of various stages, first including the use of dynamic bandwidth allocation strategies. The system will simulate bandwidth allocation mainly in three primary categories:

- Low priority: Public internet.
- Medium priority: Emergency communication.
- High priority: CCTV monitoring.

A. Simulation Environment

Mininet is a network emulator that supports SDN that will be used to model the smart city network. The numerous components of smart cities, including CCTV cameras, emergency communication tools, and public internet nodes, will be modeled on Mininet. This environment allows for the evaluation of bandwidth allocation strategies under real-time traffic flow in a controlled environment.

B. SDN Controller Configuration

The smart city network will be managed by the POX SDN controller. POX offers a versatile framework for creating bespoke network control programs. POX will be set up in this project to monitor network traffic, pinpoint bandwidth needs for various services, and dynamically distribute bandwidth depending on the importance of every service.

C. Equations and Rules

The model dynamically adjusts bandwidth based on usage and priority, ensuring efficient resource utilization and QoS compliance. The objective function and constraints are derived from a real-world scenario using SDN principles. The mathematical model for adaptive bandwidth allocation in fog computing environments is described below.

Let:

- B_{h1}, B_{h2}, B_{h3}: Allocated bandwidth for hosts $h1$, $h2$, $h3$ (in Mbps).
- U_{h1}, U_{h2}, U_{h3}: Bandwidth usage for $h1$, $h2$, $h3$ (in Mbps).
- P_{h1}, P_{h2}, P_{h3}: Priority weights for $h1$, $h2$, $h3$ respectively.
- B_h^{default}: Default bandwidth for each host.
- C: Total available bandwidth capacity (in Mbps).

The objective function is to maximize the total effective bandwidth allocation, weighted by priority:

$$\text{Maximize} \sum_{i\in\{h1,h2,h3\}} P_i \cdot B_i \quad (1)$$

The system is subject to the following constraints:

- For each host $h \in \{h1, h2, h3\}$:

$$U_h \leq B_h \quad (2)$$

- Bandwidth adjustment is triggered if usage exceeds 95% of the allocated bandwidth:

$$U_h > 0.95 \cdot B_h \rightarrow B_h = B_h + 2 \quad (3)$$

- If bandwidth usage is under control, reset to default bandwidth:

$$U_h \leq 0.95 \cdot B_h \rightarrow B_h = B_h^{\text{default}} \quad (4)$$

where the default bandwidth is:

$$B_h^{\text{default}} = \begin{cases} 10 & \text{if } h = h1 \\ 5 & \text{if } h = h2 \\ 2 & \text{if } h = h3 \end{cases} \quad (5)$$

- The total allocated bandwidth must not exceed network capacity:

$$\sum_{i\in\{h1,h2,h3\}} B_i \leq C \quad (6)$$

Equation (5) defines the default bandwidth B_h^{default} assigned to each host h. The values 10, 5, and 2 Mbps represent the minimum guaranteed bandwidth, based on the priority or requirements of each host. These values are predetermined based on empirical analysis of network demands. Equation (6) ensures that the total allocated bandwidth does not exceed the network's maximum capacity C. Here, B_i represents the allocated bandwidth for each host i, whereas C is the available network bandwidth. This constraint prevents overallocation and ensures efficient bandwidth distribution. These policies will be enforced by the SDN controller in real time, allowing for flexible bandwidth reallocation as network conditions change. These rules will distribute bandwidth depending on the following priorities:

- CCTV monitoring: Top priority to ensure a solid and continuous video feed, thus guaranteeing public safety.
- Emergency communication: Medium priority to guarantee consistent communication for police, ambulances, and fire departments in critical circumstances.
- Public internet: Lowest priority, allowing for some bandwidth reduction. Bandwidth for public internet will be dynamically adjusted based on resource availability after high- and medium-priority services are fulfilled.

D. Network Topology

The network topology is depicted in Figure 1. The network comprises three hosts ($h1$, $h2$, $h3$) connected to a central switch, each with an assigned IP address:

- Host $h1$: Designated for high-priority traffic (IP: 10.0.0.1), and it is utilized for critical services such as surveillance systems.
- Host $h2$: Handles medium-priority traffic (IP: 10.0.0.2), which includes emergency data transmission.
- Host $h3$: Manages low-priority traffic (IP: 10.0.0.3), such as general internet browsing.

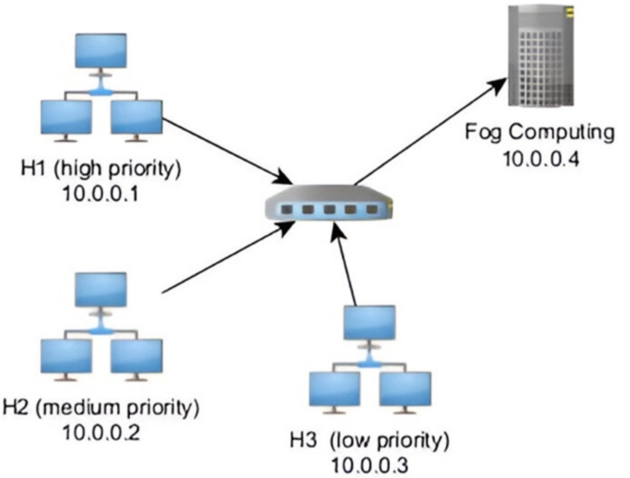


Fig. 1. Network topology with three priority levels.

The three hosts communicate with a fog computing server (IP: 10.0.0.4), which processes data at the edge of the network, ensuring efficient bandwidth allocation based on priority.

Furthermore, the three host devices are coupled to an SDN-controlled switch, with each host device representing a separate service with changing bandwidth priority:

- Host $h1$ (high priority): Represents CCTV monitoring; the network gives this maximum priority to guarantee continuous video streaming.
- Host $h2$ (medium priority): Represents emergency communication services, which are designed to maintain dependable connections during crises.
- Host $h3$ (low priority): Represents public internet connections, which have the lowest priority. The system allocates less bandwidth to lower-priority services when higher-priority services require more resources.

E. Adaptive Dynamic Bandwidth Allocation Algorithm

The algorithm presented in Algorithm 1 dynamically allocates network bandwidth in SDN based on traffic priority in an adaptive approach. Traffic is classified into high priority (e.g., CCTV monitoring), medium priority (e.g., emergency services), and low priority (e.g., internet browsing). This dynamic allocation of bandwidth is critical for ensuring optimal network performance during events, as it can adjust in real time to accommodate varying demands.

```
Algorithm 1: Adaptive dynamic bandwidth allocation
Data: $C_L, IP_{src}, Prt_{dst}, Pr_h, Pr_m, Pr_l, P$
Result: $\forall Bw \in \{P = Pr_h | P = Pr_m | P = Pr_l\}$
Initialization:
$C_a, Hb \leftarrow \{(host1_{bw} = 10, Pr_h), (host2_{bw} = 5, \mathrm{Pr}_m), (host3_{bw} = 2, \mathrm{Pr}_l)\}$, $T \leftarrow 5$,
$monitor\_traffic$, $Rec \leftarrow True$
for $C_L \in C_a$ do
    $monitor\_traffic(host1_{bw}, host2_{bw}, host3_{bw})$
end for
for $stat \in event.stats$ do
    if $Prt_{dst} = 5001$ then
        if $host1_{bw}.now > host1_{bw} \times 0.95$ then
            $host1_{bw} \mathrel{+}= 2, apply\_new\_rule$
        else
            $bandwidth\_under\_control, reset_{bw}(host1_{bw})$
        end if
    else if $Prt_{dst} = 5002$ then
        if $host2_{bw}.now > host2_{bw} \times 0.95$ then
            $host2_{bw} \mathrel{+}= 2, apply\_new\_rule$
        else
            $bandwidth\_under\_control, reset_{bw}(host2_{bw})$
        end if
    else if $Prt_{dst} = 5003$ then
        if $host3_{bw}.now > host3_{bw} \times 0.95$ then
            $host3_{bw} \mathrel{+}= 2, apply\_new\_rule$
        else
            $bandwidth\_under\_control, reset_{bw}(host3_{bw})$
        end if
    end if
end for
for $C_L \in C_a$ do
    $Pr_h \leftarrow OFm$
    if $Pr_h = (IP_{src} = \text{"10.0.0.1"} \wedge Prt_{dst} = 5001)$ then
```

```
        $C_L\ (Pr_h\ =\ 100)$
        $log.info\ ("High - Priority\ Bandwidth")$
    end if
    $Pr_m \leftarrow OFm$
    if $Pr_m\ =\ (IP_{src} = "10.0.0.2" \wedge Prt_{dst}\ =\ 5002)$ then
        $C_L\ (Pr_m\ =\ 50)$
        $log.info\ ("Medium - Priority\ Bandwidth")$
    end if
    $Pr_i \leftarrow OFm$
    if $Pr_m\ =\ (IP_{src} = "10.0.0.3" \wedge Prt_{dst}\ =\ 5003)$ then
        $C_L\ (Pr_i\ =\ 10)$
        $log.info\ ("Low - Priority\ Bandwidth")$
    end if
end for
```

During the initialization process, the algorithm starts listening for connections from the OpenFlow component on the SDN controller and sets a timer to trigger bandwidth allocation every 10 s. It enforces rules for high priority traffic (e.g., CCTV with IP 10.0.0.1 on port 5001), allocating 100% bandwidth. Medium priority traffic (e.g., emergency services with IP 10.0.0.2 on port 5002) is allocated 50%, whereas low priority traffic (e.g., browsing with IP 10.0.0.3 on port 5003) receives 10%. The OpenFlow controller applies these rules to respective connections. The launch() function registers the AdaptiveFogManagement class, enabling automatic execution. Bandwidth usage is monitored at every event. If usage exceeds 95% of the allocated bandwidth (host_bandwidth_allocated * 0.95), the bandwidth is increased by 2 Mbps. If usage is lower, the bandwidth is reset to the original allocation. This approach prioritizes critical traffic, such as CCTV and emergency services, dynamically adapting bandwidth allocation to optimize traffic flow and resource utilization.

F. Performance Evaluation

A comparative evaluation of the adaptive dynamic bandwidth allocation system with conventional static allocation methods will facilitate a determination of its performance. The main assessment benchmarks are as follows:

- Bandwidth adaptive testing: The system must be able to dynamically adjust bandwidth allocation across various priority levels without compromising network stability or performance.
- QoS: Emphasizing dependability and efficiency of bandwidth distribution for every service, QoS demonstrates the overall performance of the network, including metrics such as latency, packet lost, jitter.
- Network efficiency: The speed of data transmission within the network is crucial, especially for high-priority services like CCTV, ensuring real-time video runs smoothly without disruptions.

The evaluation will compare network performance before and after implementing adaptive dynamic bandwidth allocation. This comparison will highlight improvements in latency, QoS, and bandwidth utilization.

III. RESULTS AND DISCUSSION

A. Bandwidth Adaptive Testing

The objective of this study is to examine the protocol's ability to manage traffic based on predefined rules, with the flexibility to adapt these rules, when necessary, as opposed to the conventional protocols. To evaluate the system's adaptability in managing network load, testing is conducted in four rounds, as shown in Table I. Figure 2 illustrates the dynamic changes in bandwidth allocation over several iterations. The high-priority host ($h1$) experiences a significant increase in bandwidth at certain iterations, whereas the low-priority host ($h3$) experiences a decrease in bandwidth, reflecting an adaptive mechanism to support QoS based on priority.

TABLE I. ADAPTIVE BANDWIDTH ALLOCATION PER ROUND

Round	**$h1$ bandwidth (Mbps)**	**$h2$ bandwidth (Mbps)**	**$h3$ bandwidth (Mbps)**
Round 1	12	17	6
Round 2	32	19	2
Round 3	24	11	2
Round 4	10	21	8

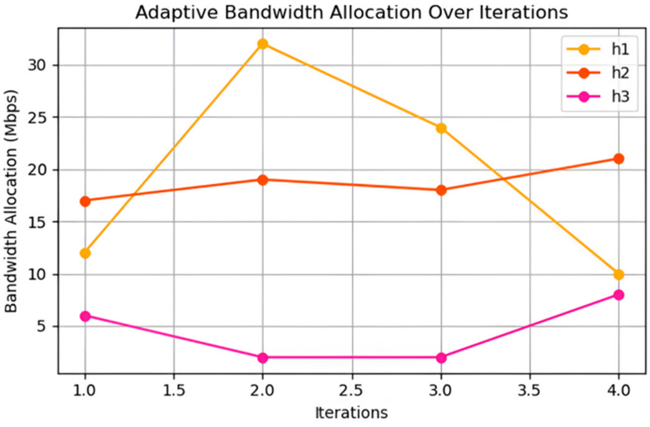


Fig. 2. Plot of adaptive bandwidth allocation per round.

B. Quality of Service

The following evaluation metrics assess network latency, defined as the duration of data transmission from its source to its destination. Figure 3 illustrates the average latency between the host and server across a ping sequence. The high-priority host ($h1$) achieves the lowest average latency (0.148 ms), demonstrating efficient bandwidth allocation for critical applications like CCTV surveillance. This evaluation highlights the ability of the adaptive bandwidth allocation system to prioritize traffic dynamically, ensuring low latency for high-priority hosts while maintaining sufficient response times for lower-priority hosts. The results confirm that the proposed adaptive bandwidth allocation method efficiently manages latency compared to traditional methods. This approach ensures optimal performance for critical applications without compromising overall network efficiency, facilitating the allocation of resources according to priority and necessity.

Figure 4 shows each host's packet transmission success status ($h1$, $h2$, $h3$) that is obtained from the command line. Each host exhibited a consistent packet transmission success rate, with a value of "1" indicating no packet loss. This outcome suggests that no packet loss occurred during any of the sequences, demonstrating the network's stability and the efficacy of the adaptive bandwidth allocation method in maintaining uninterrupted data transmission, especially for high-priority services such as CCTV ($h1$). Figure 5 presents a comparison of throughput (Mbps) and jitter (ms) for each host. The results indicate that $h1$ has the highest throughput (8.82 Mbps) and the lowest jitter (4.958 ms), thereby supporting real-time applications. In contrast, $h3$ exhibits lower throughput (1.94 Mbps) and higher jitter (51.646 ms), indicating its lower priority for general internet applications.

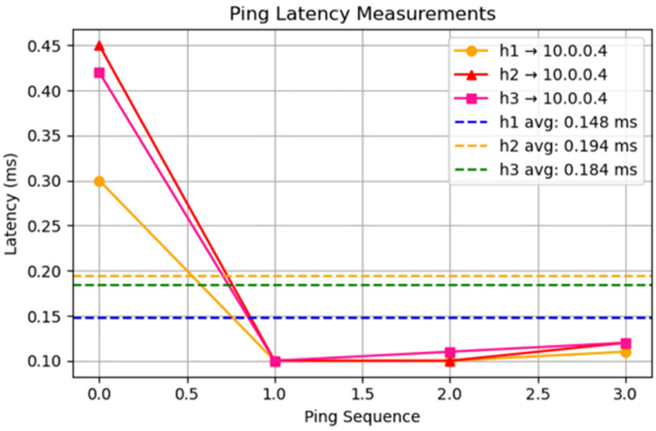


Fig. 3. Latency comparison showing minimal delay for high-priority traffic.

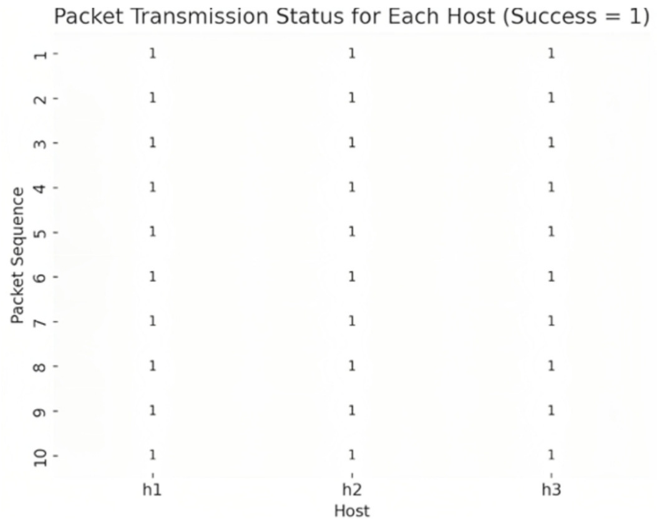


Fig. 4. Packet delivery across hosts with no data loss.

C. Network Efficiency

The objective of the test illustrated in Figure 6 is to demonstrate the superior efficiency of the proposed network model. The performance of the network is evaluated using iperf in a Mininet environment with three hosts ($h1$, $h2$, $h3$) sending data to the server on IP 10.0.0.4 through different ports (5001, 5002, and 5003). This scenario was tested several times to ensure the data were transferred correctly. The results indicate significant variations in bandwidth among the three hosts. A more thorough examination reveals that host $h1$ exhibits the best performance, with a total data transfer of 34.6 MB for 30 s and an average bandwidth of 9.65 Mbps. This result indicates that port 5001 has a higher capacity or priority than other ports, in accordance with the established regulations.

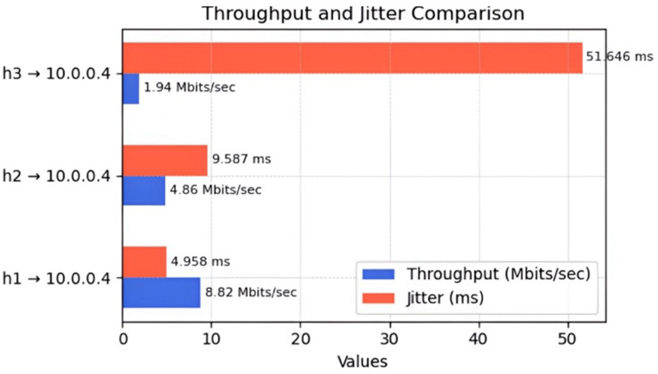
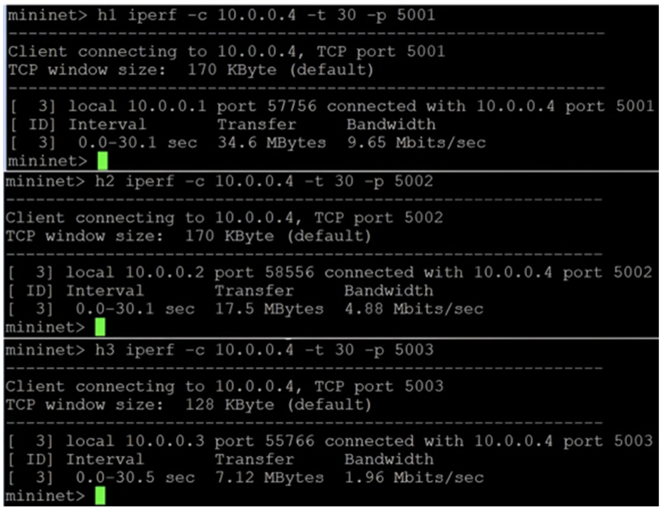


Fig. 5. Analysis of jitter and throughput illustrating the impact of traffic priority on QoS.



```
mininet> h1 iperf -c 10.0.0.4 -t 30 -p 5001
------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size:  170 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 57756 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-30.1 sec  34.6 MBytes  9.65 Mbits/sec
mininet> 
mininet> h2 iperf -c 10.0.0.4 -t 30 -p 5002
------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5002
TCP window size:  170 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.2 port 58556 connected with 10.0.0.4 port 5002
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-30.1 sec  17.5 MBytes  4.88 Mbits/sec
mininet> 
mininet> h3 iperf -c 10.0.0.4 -t 30 -p 5003
------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5003
TCP window size:  128 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.3 port 55766 connected with 10.0.0.4 port 5003
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-30.5 sec  7.12 MBytes  1.96 Mbits/sec
mininet> 
```

Fig. 6. Bandwidth allocation results for network efficiency testing.

Meanwhile, host $h2$, utilizing port 5002, recorded a total data transfer of 17.5 MB with an average bandwidth of 4.88 Mbps. Its performance decreased by almost half of the host $h1$, which could be attributed to resource sharing or lower priority settings on port 5002. Conversely, host $h3$, connected via port 5003, exhibited the lowest performance, with a total data transfer of only 7.12 MB and an average bandwidth of 1.96 Mbps. These results indicate significant limitations, or what we call a New York efficiency, both in terms of bandwidth allocation and potential bottlenecks in the network. Several factors, including suboptimal priority scheduling or limited resource allocation on the relevant port, could result in a substantial performance decrease in this port. To address this challenge, we have proposed an adaptive network capable of adjusting its rules in accordance with the specific circumstances. These results also reveal uneven network performance, depending on the port used. This finding suggests that all ports adhere to the established rule.

IV. CONCLUSION

This research demonstrates the effectiveness of adaptive dynamic bandwidth allocation using Software-Defined Networking (SDN) within smart city environments. The proposed system leverages the benefits of SDN's centralized control and programmability to prioritize bandwidth allocation for high-demand services, ensuring optimal network performance when necessary. The experimental results validate the system's ability to dynamically adjust bandwidth based on real-time traffic conditions and service priorities. It has been demonstrated that high-priority services, such as Closed-Circuit Television (CCTV) surveillance, consistently achieved low latency, high throughput, and minimal jitter, ensuring reliability for critical applications. Conversely, lower-priority services, such as public internet access, maintained sufficient bandwidth without compromising overall network efficiency. Furthermore, the proposed algorithm effectively maintained stability in real-time testing by dynamically reallocating bandwidth during times of heavy traffic demand, thus reducing packet loss and improving Quality of Service (QoS).

The proposed approach demonstrates significant advancements in latency, throughput, and resource utilization compared to conventional static allocation techniques. These results emphasize the scalability and efficiency of SDN-based adaptive bandwidth distribution as a solution for the dynamic and complex demands of smart city networks. Future research could extend this methodology by incorporating machine learning techniques to increase flexibility and enable predictive traffic management.

ACKNOWLEDGMENT

The authors would like to express their gratitude to the Department of Computer Science and Information Engineering, NTUST, for providing the necessary support and facilities for the completion of this project. Special thanks are extended to Professor Zelalem Legese Hailemariam for his invaluable guidance, constructive feedback, and encouragement throughout this research. His expertise and insights have significantly contributed to the quality and depth of this work.

REFERENCES

[1] Z. E. Ahmed, A. A. Hashim, R. A. Saeed, and M. M. Saeed, "Enhancing Smart City Mobility Using Software Defined Networks," in *2024 9th International Conference on Mechatronics Engineering*, Kuala Lumpur, Malaysia, 2024, pp. 299–303, https://doi.org/10.1109/ICOM61675.2024.10652267.

[2] I. Alam *et al.*, "A Survey of Network Virtualization Techniques for Internet of Things Using SDN and NFV," *ACM Computing Surveys*, vol. 53, no. 2, Apr. 2020, Art. no. 35, https://doi.org/10.1145/3379444.

[3] A. A. Ateya, A. Muthanna, A. Koucheryavy, Y. Maleh, and A. A. A. El-Latif, "Energy efficient offloading scheme for MEC-based augmented reality system," *Cluster Computing*, vol. 26, no. 1, pp. 789–806, Feb. 2023, https://doi.org/10.1007/s10586-022-03914-7.

[4] 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, Sep. 2022, Art. no. 109133, https://doi.org/10.1016/j.comnet.2022.109133.

[5] V. Demiroglou, S. Skaperas, L. Mamatas, and V. Tsaoussidis, "Adaptive Multiprotocol Communication in Smart City Networks," *IEEE Internet of Things Journal*, vol. 11, no. 11, pp. 20499–20513, Jun. 2024, https://doi.org/10.1109/JIOT.2024.3372624.

[6] U. Ghosh, P. Chatterjee, S. Shetty, and R. Datta, "An SDN-IoT-based Framework for Future Smart Cities: Addressing Perspective." arXiv, Jul. 22, 2020, https://doi.org/10.48550/arXiv.2007.11536.

[7] D. Goltzsche *et al.*, "EndBox: Scalable Middlebox Functions Using Client-Side Trusted Execution," in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, Luxembourg, Luxembourg, 2018, pp. 386–397, https://doi.org/10.1109/DSN.2018.00048.

[8] C. T. E. R. Hewage, A. Ahmad, T. Mallikarachchi, N. Barman, and M. G. Martini, "Measuring, Modeling and Integrating Time-Varying Video Quality in End-to-End Multimedia Service Delivery: A Review and Open Challenges," *IEEE Access*, vol. 10, pp. 60267–60293, 2022, https://doi.org/10.1109/ACCESS.2022.3180491.

[9] J.-C. Kao, G.-H. Ma, C.-Y. Lee, C.-F. Kuo, and J.-H. Hong, "Load-Balancing and Prudent Deployment of VNFs for Heterogeneous Multicore Systems," in *2024 IEEE Wireless Communications and Networking Conference*, Dubai, United Arab Emirates, 2024, pp. 1–6, https://doi.org/10.1109/WCNC57260.2024.10571009.

[10] A. Ben Letaifa, "Real Time ML-Based QoE Adaptive Approach in SDN Context for HTTP Video Services," *Wireless Personal Communications*, vol. 103, no. 3, pp. 2633–2656, Dec. 2018, https://doi.org/10.1007/s11277-018-5952-6.

[11] A. Manzalini *et al.*, *Software-Defined Networks for Future Networks and Services: Main Technical Challenges and Business Implications.* New York, NY, USA: IEEE, 2014.

[12] N. J. Mocelin Júnior and A. Fiorese, "FLOWPRI-SDN: A Framework for Bandwidth Management for Prioritary Data Flows Applied to a Smart City Scenario," in *Proceedings of the 37th International Conference on Advanced Information Networking and Applications (AINA-2023), Volume 1*, Juiz de Fora, Brazil, 2023, pp. 346–357, https://doi.org/10.1007/978-3-031-29056-5_31.

[13] A. Nain, S. Sheikh, M. Shahid, and R. Malik, "Resource optimization in edge and SDN-based edge computing: a comprehensive study," *Cluster Computing*, vol. 27, no. 5, pp. 5517–5545, Aug. 2024, https://doi.org/10.1007/s10586-023-04256-8.

[14] F. Pizzato, D. Bringhenti, R. Sisto, and F. Valenza, "Automatic and optimized firewall reconfiguration," in *NOMS 2024-2024 IEEE Network Operations and Management Symposium*, Seoul, South Korea, 2024, pp. 1–9, https://doi.org/10.1109/NOMS59830.2024.10575212.

[15] A. Qadeer, M. J. Lee, and K. Tsukamoto, "Flow-Level Dynamic Bandwidth Allocation in SDN-Enabled Edge Cloud using Heuristic Reinforcement Learning," in *2021 8th International Conference on Future Internet of Things and Cloud*, Rome, Italy, 2021, pp. 1–10, https://doi.org/10.1109/FiCloud49777.2021.00009.

[16] A. Rahman *et al.*, "SDN–IoT empowered intelligent framework for industry 4.0 applications during COVID-19 pandemic," *Cluster Computing*, vol. 25, no. 4, pp. 2351–2368, Aug. 2022, https://doi.org/10.1007/s10586-021-03367-4.

[17] C. Rametta, G. Baldoni, A. Lombardo, S. Micalizzi, and A. Vassallo, "S6: a Smart, Social and SDN-based Surveillance System for Smart-cities," *Procedia Computer Science*, vol. 110, pp. 361–368, Jan. 2017, https://doi.org/10.1016/j.procs.2017.06.078.

[18] G. Singh and G. Kaur, "Design and analysis of novel SDN-controlled dynamically reconfigurable TDM-DWDM-based optical network for smart cities," *Photonic Network Communications*, vol. 47, no. 2, pp. 57–78, Apr. 2024, https://doi.org/10.1007/s11107-023-01012-1.

[19] M. D. Tache (Ungureanu), O. Păscuțoiu, and E. Borcoci, "Optimization Algorithms in SDN: Routing, Load Balancing, and Delay Optimization," *Applied Sciences*, vol. 14, no. 14, Jul. 2024, Art. no. 5967, https://doi.org/10.3390/app14145967.

[20] Y. Djeldjeli and M. Zoubir, "CP-SDN: A New Approach for the Control Operation of 5G Mobile Networks to Improve QoS," *Engineering,*

Technology & Applied Science Research, vol. 11, no. 2, pp. 6857–6863, Apr. 2021, https://doi.org/10.48084/etasr.4016.

[21] M. H. H. Khairi, S. H. S. Ariffin, N. M. A. Latiff, A. S. Abdullah, and M. K. Hassan, "A Review of Anomaly Detection Techniques and Distributed Denial of Service (DDoS) on Software Defined Network (SDN)," *Engineering, Technology & Applied Science Research*, vol. 8, no. 2, pp. 2724–2730, Apr. 2018, https://doi.org/10.48084/etasr.1840.

[22] M. K. Hassan *et al.*, "DLVisor: Dynamic Learning Hypervisor for Software Defined Network," *IEEE Access*, vol. 11, pp. 84144–84167, 2023, https://doi.org/10.1109/ACCESS.2023.3302266.