Dynamic Reinforcement Learning-Driven Digital Twin for Optimised Multimedia Traffic Management in B5G SDN Core Networks

Ahmed Al-Jawad[1], Mahnoor Yaqoob[2], Purav Shah[2], Huan Nguyen[2], Ramona Trestian[2], and Ioan-Sorin Comşa[3]

[1]Department of Design Engineering and Mathematics,
Middlesex University London, NW4 4BT, Hendon, London, U.K.

[2]London Digital Twin Research Centre, Middlesex University London, NW4 4BT, Hendon, London, U.K.

[3]Institute for Research in Open-, Distance- and eLearning,
Swiss Distance University of Applied Sciences, CH-3900, Brig, Switzerland.

E-mails: {AA3512, MY365}@live.mdx.ac.uk, {p.shah, h.nguyen, r.trestian}@mdx.ac.uk,
ioan-sorin.comsa@ffhs.ch

***Abstract*—The rapid growth of multimedia applications and the rising expectations for an enhanced Quality of Experience (QoE) among users have emphasised the need for innovative approaches to ensure efficient delivery of video content in 5G and Beyond 5G (B5G) networks. Machine Learning (ML) techniques are increasingly being explored to address these challenges by enabling intelligent traffic management and QoE optimisation. Within this landscape, Software-Defined Networking (SDN) plays a pivotal role as a facilitator of dynamic resource allocation and QoE-centric network management. This paper introduces AIMTWIN, a reinforcement learning (RL)-driven Digital Twin framework designed to optimise multimedia traffic management in B5G SDN environments. AIMTWIN integrates real-time telemetry from physical networks with a dynamic virtual model to provide adaptive and efficient traffic routing. By prioritising static paths for QoS-critical multimedia flows and dynamically managing background traffic, the framework delivers superior network performance and user satisfaction. Experimental results on small and medium scale topologies highlight AIMTWIN's ability to achieve consistently *Excellent* QoE compared to state-of-the-art routing methods, positioning it as a scalable and robust solution for next-generation networks.**

***Index Terms*—Reinforcement Learning, Multimedia, Beyond 5G, Software-Defined Networks, Digital Twin, QoS, QoE.**

I. Introduction

The rapid evolution of multimedia services has been a significant driver for advancements in Beyond 5G (B5G) networks. These networks must support diverse service classes such as ultra-high-definition video streaming, real-time interactive applications, and emerging technologies like augmented and virtual reality (AR/VR), all of which demand stringent Quality of Service (QoS) requirements. Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency Communications (URLLC), and Massive Machine-Type Communications (mMTC) represent critical service paradigms requiring robust network infrastructures [1], [2].

Multimedia traffic management has grown increasingly complex due to the heterogeneous nature of these service classes. Traditional network architecture struggles to handle the dynamic and stringent requirements of such traffic. This challenge has been addressed through the adoption of Software-Defined Networking (SDN) and Digital Twin (DT) technologies, which collectively enable real-time network adaptation and predictive management [3], [4].

SDN decouples the control and data planes, allowing centralised traffic management and dynamic resource allocation. This programmability is particularly beneficial for handling high-bandwidth, latency-sensitive applications, as SDN enables network slicing to meet the specific needs of multimedia service [4], [5]. Furthermore, integration with Network Function Virtualisation (NFV) enhances scalability and reduces the operational complexity associated with managing Virtualised Network Functions (VNFs) [6].

Digital Twin technology offers a sophisticated approach to network modelling by creating a virtual counterpart of the physical network. This virtual model, equipped with real-time bi-directional communication, enables predictive analytics, scenario testing, and optimised decision-making without disrupting the live network [4], [7]. The B5G DTs provide a controlled environment to simulate and optimise traffic scenarios, predict failures, and enhance QoS adherence. The integration of Artificial Intelligence (AI) within DTs has further revolutionised traffic management by facilitating adaptive learning and resource optimisation [2], [3].

This paper introduces AIMTWIN (AI Multimedia TWIN), a dynamic Reinforcement Learning (RL)-driven DT framework for optimised multimedia traffic management in B5G SDN core networks. AIMTWIN leverages reinforcement learning to dynamically select the most suitable routing algorithms, effectively managing background traffic while ensuring adherence to Service-Level Agreement (SLA) objectives for multimedia applications. By addressing the dual challenges of traffic heterogeneity and stringent QoS requirements, AIMTWIN paves the way for resilient, efficient, and scalable network operations in the era of Beyond 5G.

II. Related Works

The growing complexity of network environments, driven by the explosion of multimedia traffic, has necessitated the development of advanced technologies such as SDN, DTs, and

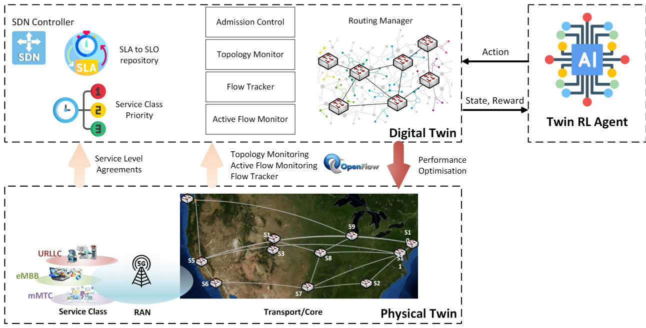

Fig. 1: AIMTWIN Framework

AI. These innovations aim to enhance resource management, ensure high QoS, and support the heterogeneous requirements of B5G networks.

DT have emerged as a transformative technology for B5G and 6G systems. A DT of the Network (DTN) provides a real-time virtual representation of a network, enabling operators to simulate, monitor, and optimise network behaviour without impacting live operations. For instance, DTNs can predict network failures, optimise resource allocations, and improve service provisioning by analysing both historical and real-time data [8], [9]. Lin et al., [8] discuss the theoretical underpinnings and design of DTNs for 6G systems, while Mozo et al. [9] explores the B5GEMINI project, which integrates AI and DTNs for resource optimisation in B5G networks. DTNs are particularly advantageous in managing the dynamic and heterogeneous demands of multimedia traffic by leveraging AI and Machine Learning (ML) models for predictive analysis and decision-making [10], [11].

The integration of SDN with DTNs is pivotal for enhancing network programmability and dynamic resource management. SDN's centralised control architecture simplifies the collection of network telemetry data, which can be fed into DT models to predict and address potential performance bottlenecks. AI-driven frameworks, like those incorporating reinforcement learning, enable adaptive control of network resources to maintain the SLAs for multimedia services [11], [12]. For example, Boffetti et al. [11] discuss the design of AI-based DTNs for multimedia service provisioning. The authors explore the Quality of Experience (QoE)/QoS prediction using AI models like Long Short-Term Memory (LSTM) and Deep RL for optimal resource allocation in multimedia networks. RL is increasingly being utilised in DT-enhanced networks for tasks like network slicing, traffic engineering, and SLA management. RL algorithms interact with DT-based virtual environments to learn optimal policies without directly affecting the physical network. The use of RL has demonstrated improved efficiency in scheduling problems [13], [14], and when used in DTNs in resource allocation problems for network slicing, the dynamic traffic demands are met by pre-training the RL models in virtual environments [15], [16].

The integration of DTNs, SDN, and AI represents a cutting-edge approach to multimedia traffic management in B5G. These technologies collectively address the challenges of dynamic traffic patterns, stringent QoS requirements, and efficient resource utilisation, laying the groundwork for scalable and resilient future network infrastructures.

In our prior research, we extensively applied RL algorithms to dynamically schedule mobile users across time and frequency domains, aiming to enhance throughput, minimize delay and packet loss, while ensuring fairness among users [17], [18]. When applied to SDNs, RL-based framework were used to dynamically select the most suitable routing algorithm for QoS-based traffic flows in an SDN environment [19], [20]. While this approach improved QoS provisioning, it treated the routing strategy for background traffic as static, which inadvertently led to frequent interruptions of QoS-based flows due to the periodic re-routing decisions for video traffic. These interruptions significantly impacted the users' QoE.

In contrast to the earlier framework [21], AIMTWIN introduces the concept of a Digital Twin (DT) to enable more dynamic and intelligent traffic management. Unlike prior approaches, AIMTWIN ensures that video flows remain static, avoiding the disruptions caused by re-routing, while dynamically adapting the routing of background traffic to optimise overall network performance and QoS objectives. This novel methodology enhances network stability and ensures higher levels of user satisfaction, addressing the limitations of previous RL-based implementations.

III. Proposed AIMTWIN Framework

The proposed AIMTWIN framework is illustrated in Fig. 1 and consists of the Physical Twin (PT), the DT and the Twin RL Agent. The PT represents the real-world 5G SDN network, encompassing the Radio Access Network (RAN), transport/core network, and the service classes, including eMBB, URLLC, and mMTC. This physical layer handles live traffic flows and captures real-time telemetry, including topology information, active flows, and performance metrics. The network employs OpenFlow switches to enable programmability and interoperability with the SDN controller. The DT mirrors the PT and provides a virtual environment for real-time simulation, performance optimisation, and routing decisions. This twin comprises several modules: *Topology Monitor* – continuously observes the network topology and logs updates; *Flow Tracker* - tracks active and inactive flows, recording statistics such as throughput, latency, and packet loss; *Admission Control* - decides whether to accept or reject new traffic flows based on resource availability and Service Level Objectives (SLOs); *Routing Manager* - dynamically adjusts routing strategies by interacting with the TWIN RL Agent to optimise background traffic flows.

The TWIN RL Agent operates within the DT, learning optimal policies to route background traffic based on the state of the network. The RL model considers state variables such as link utilisation, active flow metrics, and QoS violations to decide actions (e.g., routing algorithm selection). These actions are fed back to the DT for evaluation, creating a feedback loop for continuous learning and optimisation. The AIMTWIN framework workflow is as follows: (1) The SDN controller collects real-time telemetry from the PT and shares it with the DT. (2) The DT processes the data and uses the TWIN RL Agent to simulate and evaluate potential routing strategies. (3) The TWIN RL Agent selects the optimal routing strategy, which the Routing Manager applies to the background traffic flows. (4) Performance feedback from the PT is sent back to the DT to refine future routing decisions.

IV. System Model

The system model for the AIMTWIN framework is based on the dynamic management of background traffic flows in a B5G SDN core network, RL within a DT environment. Specifically, the objective is to improve the QoS and QoE of multimedia flows by adaptively re-routing the background traffic with different algorithms.

A. Problem Formulation

The B5G SDN core network is modeled as a graph $G(\mathcal{N}, \mathcal{L})$, where $\mathcal{N} = \{n_1, n_2, ..., n_N\}$ represents the set of SDN switches with N as the total number of nodes. The set of links connecting these nodes is denoted by $\mathcal{L} = \{l_1, l_2, ..., l_L\}$, with L being the total number of possible links within the network graph G. Each link $l \in \mathcal{L}$ is characterized by a capacity Ω_l and a remaining bandwidth bw_l, which depends on the current routing state. Based on the links $l \in \mathcal{L}$, we form a set of paths $\mathcal{P} = \{p_1, p_2, ..., p_P\}$, where P is the total number of paths that can be constructed within graph G. Here, each path $p \in \mathcal{P}$ is a subset of $\mathcal{L}$. Let us consider $\mathcal{C} = \{c_1, c_2, ..., c_c\}$ the set of traffic classes that are active through graph G, with C the maximum number of classes. Each of these classes has a particular set of flows $\mathcal{F}_c = \{f_{c,1}, f_{c,2}, ..., f_{c,F_c}\}$ that needs a routing or rerouting decision at certain time steps t, where $F_c(t)$ is the number of flows belonging to service class $c \in \mathcal{C}$ following a certain routing decision at time t. Once the routing algorithm decides the path to follow for flow $f \in \mathcal{F}_c$ of class $c \in \mathcal{C}$, the remaining bandwidth of the link $l \in \mathcal{L}$ is computed as:

$$bw_l = \Omega_l - \sum_{c \in \mathcal{C}} \sum_{f \in \mathcal{F}_c} \sum_{p \in \mathcal{P}} x_{c,f,p} \cdot y_{p,l} \cdot \rho_{c,f,p}, \quad \forall l \in \mathcal{L}, \quad (1)$$

where $y_{p,l} = 1$ if link $l \in \mathcal{L}$ is part of path $p \in \mathcal{P}$, otherwise $y_{p,l} = 0$; $x_{c,f,p} = 1$ if flow $f \in \mathcal{F}_c(t)$ traverses path $p \in \mathcal{P}$, otherwise $x_{c,f,p} = 0$; $\rho_{c,f,p}$ represents the bit rate (throughput) of flow $f \in \mathcal{F}_c(t)$ belonging to class $c \in \mathcal{C}$ when routed through path $p \in \mathcal{P}$.

The objective of a routing algorithm is to determine the optimal paths for all active flows per service class in order to: *a)* avoid congestion on network links and minimize rejection rate of traffic class; *b)* meet the SLA requirements in terms of minimum throughput of each flow; and *c)* reduce packet loss for each flow in each class. Different routing strategies are applied to satisfy heterogeneous QoS constraints while adapting to network conditions. This transforms the routing problem into a complex multi-objective optimisation problem, which can be written as presented by (2). Here, the variables are explained as follows: $\rho_{c,f,p}$ is the throughput of flow f corresponding to traffic class c when routed on path p; $\theta_{c,f,p}$ is the packet loss rate of flow f of class c when routed on path p; η_c represents the rejection rate of class c as a result of routing decisions. The SLA requirements are specified for each of these variables, as follows: $\rho^{min}_{c,f,p}$ is the minimum throughput requirement, $\theta^{max}_{c,f,p}$ the maximum requirement for packet loss, and η^{max}_c is the maximum allowable rejection rate in class c.

$$\max_{x,y} \sum_{c \in \mathcal{C}} \sum_{f \in \mathcal{F}_c} \sum_{p \in \mathcal{P}} \sum_{l \in \mathcal{L}} (\tau_1 \cdot \rho_{c,f,p} - \tau_2 \cdot \theta_{c,f,p}) \cdot x_{c,f,p}(t) \cdot \cdot y_{p,l}(t) - \tau_3 \cdot \eta_c, \quad (2)$$

$s.t.:$

$$bw_l \geq 0, \forall l \in \mathcal{L} \quad (2.a)$$

$$y_{p,l} \in \{0,1\}, \quad \forall p \in \mathcal{P}, \forall l \in \mathcal{L}, \quad (2.b)$$

$$x_{c,f,p} \in \{0,1\}, \quad \forall c \in \mathcal{C}, f \in \mathcal{F}_c, p \in \mathcal{P}, \quad (2.c)$$

$$\sum\nolimits_{l \in \mathcal{L}} y_{p,l} \leq L, \quad \forall p \in \mathcal{P}, \quad (2.d)$$

$$\sum\nolimits_{c \in \mathcal{C}} \sum\nolimits_{f \in \mathcal{F}_c} \sum\nolimits_{p \in \mathcal{P}} x_{c,f,p} \leq P, \quad (2.e)$$

$$x_{c,f,p} \cdot \rho_{c,f,p} \leq \Omega_{max}, \quad \forall c \in \mathcal{C}, f \in \mathcal{F}_c, p \in \mathcal{P}, \quad (2.f)$$

$$x_{c,f,p} \cdot \rho_{c,f,p} \geq \rho^{min}_{c,f}, \quad \forall c \in \mathcal{C}, f \in \mathcal{F}_c, p \in \mathcal{P}, \quad (2.g)$$

$$x_{c,f,p} \cdot \theta_{c,f,p} \leq \theta^{max}_{c,f}, \quad \forall c \in \mathcal{C}, f \in \mathcal{F}_c, p \in \mathcal{P}, \quad (2.h)$$

$$\eta_c \leq \eta^{max}_c, \quad \forall c \in \mathcal{C}. \quad (2.i)$$

In multi-objective optimisation, it is essential to normalize objectives to ensure each contributes appropriately to the overall goal. A common approach is to assign weights to each objective, often based on their respective SLA requirements. For instance, setting weights as the inverses of these requirements - such as $\tau_1 = 1/\rho^{min}_{c,f}, \tau_2 = 1/\theta^{max}_{c,f}, \tau_3 = 1/\eta^{max}_c$ - effectively normalizes the objectives.

In the proposed optimisation problem, the routing algorithm specifies the variables $x_{c,f,p}$ and $y_{p,l}$ for each active flow $f \in \mathcal{F}_c$ of class $c \in \mathcal{C}$, ensuring a certain degree of SLA satisfaction in terms of their requirements. Constraints (2.a) requires that each link $l \in \mathcal{L}$ should not be congested. Constraints (2.b) and (2.d) preselects the links for each possible path $p \in \mathcal{P}$. Constraints (2.c) and (2.e) indicates that the number of possible paths should not be greater than the maximum value P when routing all flows from all traffic classes. The throughput of each flow in each class should not exceed the maximum capacity as denoted by constraints (2.f). Finally, the constraints (2.g) - (2.i) aim to respect the SLA requirements in terms of throughput, packet loss and rejection rate, respectively, in each service class $c \in \mathcal{C}$.

B. TWIN RL-Based Solution

The proposed solution addresses the optimisation problem from (2) by selecting the most suitable algorithm to reroute traffic flows, ensuring adherence to SLA requirements in each class. Given the diverse nature of traffic types with varying and stringent QoS demands, we divided the service classes $\mathcal{C}$ into two classes: *a)* QoS-Based Traffic Flows ($\mathcal{C}_{qos}$) that includes multimedia services such as live HD streaming, which require higher bandwidth and lower latency; *b)* Background Traffic Flows ($\mathcal{C}_{bkg}$) that encompasses services like web browsing (HTTP), file transfers (FTP), and buffered SD videos which have less stringent QoS requirements. In the proposed framework, background traffic is dynamically rerouted by selecting appropriate routing algorithms in each iteration, while maintaining a consistent routing strategy for the QoS traffic class. This approach optimises rerouting strategies for background traffic to better accommodate services with more stringent QoS requirements, such as live HD multimedia streaming.

AIMTWIN utilizes Q-learning to dynamically select routing algorithms for background traffic, ensuring that high-priority QoS traffic receives the necessary resources and optimal paths, thereby maintaining service quality and network efficiency. At each time step t, the system observes the current network state $S_t \in \mathcal{S}$, which reflects how well the QoS traffic flows meet their SLA requirements. Based on this state, the Q-learning algorithm selects an action $A_t \in \mathcal{A}$, indicating to the Routing Manager which routing algorithm to apply for the background traffic. After executing this action, the system receives a reward $R_{t+1}(S_t, A_t)$, representing the QoS revenue after rerouting the background traffic using algorithm A_t in QoS traffic state S_t. Both the state space ($\mathcal{S}$) and action space ($\mathcal{A}$) are discrete, allowing the accumulated Q-values to be stored in a tabular form for each state-action pair. The Q-learning algorithm iteratively processes tuples of $(S_t, A_t, R_{t+1}, S_{t+1})$ over a finite number of iterations, exploring the entire state and action spaces multiple times to learn optimal policies for selecting appropriate routing algorithms. The components used by the AIMTWIN framework are:

1) States: In each iteration, the system observes a discrete network state $S_t \in \mathcal{S}$, focusing exclusively on QoS traffic. This state is represented by the vector:

$$S_t = [\mathbf{T}, \mathbf{L}, \mathbf{P}_t, \Theta_t, \mathbf{H}_t], \tag{3}$$

where: $\mathbf{T}$ represents the type of network topology used during the training of Q-values; $\mathbf{L}$ indicates the traffic load for QoS traffic, with discrete values such as $\mathbf{L} = 0$ for low, $\mathbf{L} = 1$ for medium, and $\mathbf{L} = 2$ for high traffic load; $\mathbf{P}_t$ is a binary variable indicating whether the throughput constraints are met for QoS traffic at time step t, specifically, $\mathbf{P}_t = 1$ when all constraints in (2.g) are met, and $\mathbf{P}_t = 0$, otherwise; Θ_t a binary variable indicating whether the packet loss constraints are met for QoS traffic at time t, specifically, $\Theta_t = 1$ when all constraints in (2.h) are met, and $\Theta_t = 0$, otherwise; $\mathbf{H}_t$ a binary variable indicating whether the rejection constraints are met for QoS traffic classes at time t, specifically, $\mathbf{H}_t = 1$ when all constraints in (2.i) are met, and $\mathbf{H}_t = 0$, otherwise. The size of $\mathcal{S}$ would be 9, when training the algorithm separately for each combination of topology and traffic load.

2) Actions: are decided in terms of the routing algorithm for the background traffic at each time step as $A_t \in \mathcal{A} = \{A_1, A_2, A_3, A_4\}$, where: A_1 is the Minimum Hop Algorithm (MHA), A_2 is the Widest Shortest Path (WSP), A_3 is the Shortest Widest Path (SWP) and A_4 is the Minimum Interference Routing Algorithm (MIRA) algorithm. While the background traffic is rerouted by dynamically selecting a different algorithm by the Q-learning, the QoS traffic is using a constant routing scheme.

3) Rewards: are designed to optimise traffic management for different service classes. We implement a reward function at the level of each traffic class $c \in \mathcal{C}$, as follows:

$$R_c = w_\rho \cdot R_{c,\rho} + w_\theta \cdot R_{c,\theta} + w_\eta \cdot R_{c,\eta}, \tag{4}$$

where $R_{c,\rho}, R_{c,\theta}, R_{c,\eta}$ represent the rewards associated with the SLA requirements for throughput, packet loss, and rejection rate, respectively, and calculated as detailed in [20]. The weights $\{w_\rho, w_\theta, w_\eta\}$ reflect the relative importance of each SLA objective, and typically, they are set to equal values. Then, the total reward is calculated as follows:

$$R = \sum\nolimits_{c \in \mathcal{C}} w_c \cdot R_c, \tag{5}$$

where w_c ensures prioritisation between QoS and background traffic. In this study, the AIMTWIN framework's reward function is designed to optimise traffic management across four traffic classes $\mathcal{C} = \{c_1 : HD, c_2 : SD, c_3 : FTP, c_4 : HTTP\}$, where $c_1 \in \mathcal{C}_{qos}$ represents the live HD multimedia corresponding to QoS category, while $\{c_2, c_3, c_4\} \in \mathcal{C}_{bkg}$ constitute the background traffic classes. The weights represent the priority of each class, with the values set to $w_{HD} = 0.63$, $w_{SD} = 0.19$, $w_{FTP} = 0.09$, and $w_{HTTP} = 0.09$, respectively, ensuring that their sum equals 1.

4) Q-values: are associated with each state-action pair and updated upon each visit using the following formula:

$$Q^{new}(S_t, A_t) = (1 - \alpha) \cdot Q^{old}(S_t, A_t) + \alpha \cdot \Big[R_{t+1} + \gamma \cdot max_{a' \in \mathcal{A}} Q^{old}(S_{t+1}, a')\Big], \tag{6}$$

where α is the learning rate, γ is the discount factor, Q^{old} are the old Q-values and Q^{new} are the updated values. In this framework, we consider as time step the network trial number with 1500 seconds run time per each trial. In train phase we updated the Q-values for a number of 60 trials, using $\epsilon-$greedy policy to choose actions between states with $\alpha = 0.1$ and $\gamma = 0.9$ [21]. In test phase, the trained Q-table decides the action to be applied in each network state.

TABLE I: PSNR to MOS Mapping [22]

PSNR [dB]	MOS
≥ 45	5 (Excellent)
≥ 33 & < 45	4 (Good)
≥ 27.4 & < 33	3 (Fair)
≥ 18.7 & < 27.4	2 (Poor)
< 18.7	1 (Bad)

TABLE II: QoS Requirements [21]

QoS-based Traffic Class	ρ^{min}	θ^{max}	η^{max}
Live HD Video	658 Kb/s	1%	25%
Background Traffic Class	ρ^{min}	θ^{max}	η^{max}
Buffered SD Video	279 Kb/s	2%	35%
HTTP	14 Kb/s	0%	35%
FTP	180 Kb/s	0%	35%

V. Experimental Settings and Evaluations

A. Experimental Setup

The performance of the proposed AIMTWIN framework was evaluated under an experimental setup hosted on OpenStack, that mirrors realistic network conditions. One virtual server was used for the DT environment integrating the SDN controller (Floodlight) and the TWIN RL algorithm, and another virtual server was used to emulate the Physical Twin

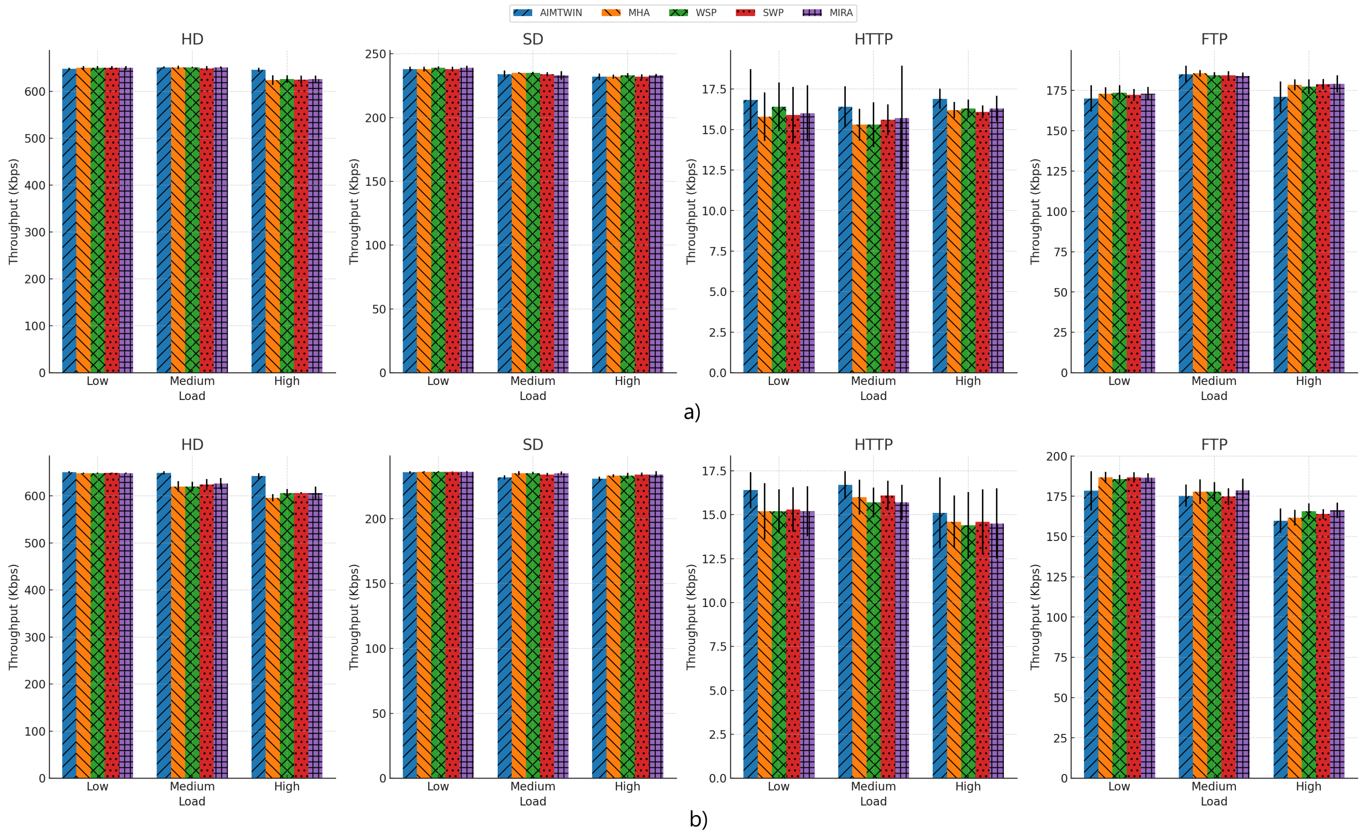

Fig. 2: Throughput Results for Each Traffic Type Considering Different Traffic Loads Over a) GetNet and b) Sprint Topologies

(PT) by running the Mininet test-bench that emulates the data plane of the SDN network.

The experiments use two distinct network topologies from Internet Topology Zoo [23]: Sprint, a middle scale topology consisting of 11 nodes and 18 links, and GetNet, a small-scale topology, consisting of 7 nodes and 8 links. SDN Open vSwitches were used to replace the network nodes.

Traffic generation was performed with the VLC Media Player with a Constant Bit Rate (CBR) encoder for HD and SD video traffic while Ostinato traffic generator was used for FTP and HTTP traffic. The QoS-based traffic class consists of live HD multimedia streaming, characterised by, 1280x720 pixels resolution, 24fps frame rate, 665Kbps average bit rate and 5 minutes duration. The background traffic consists of buffered SD video streaming with 640x360 pixels resolution, 24fps, 285Kbps and 5 minutes duration, while FTP and HTTP are modelled using standard benchmarks for data and latency characteristics. The WSP algorithm is employed as the static routing strategy for live HD multimedia streaming to ensure high QoS and prevent interruptions. Background traffic flows employ a dynamic routing decision made by the TWIN RL Agent, which selects from MHA, WSP, SWP, and MIRA.

The experiments evaluate the network under three load levels, such as: Low (50% of Ω_l), Medium (75% of Ω_l), High (100% of Ω_l). The performance of AIMTWIN is compared against other state-of-the-art routing algorithms from the literature, such as MHA, WSP, SWP, and MIRA. The evaluation is done based on throughput, rejection rate, Peak Signal-to-Noise Ratio (PSNR) and Mean Opinion Score (MOS). The PSNR to MOS mapping is done as per Table I [22], while Table II [21] lists the QoS requirements for each traffic class.

B. Performance Evaluation

The results represent the average of five simulation runs per scenario, each lasting 1500 seconds. Consistent experimental conditions were maintained across all scenarios to ensure a fair comparison of the different solutions. Figures 2.a and 2.b present the throughput results for each scheme across different traffic classes, varying traffic loads, and two network topologies, GetNet and Sprint, respectively. Under the small-scale GetNet topology, with low to medium traffic loads, the throughput performance remains comparable across all solutions. However, AIMTWIN demonstrates a more effective balance in meeting the diverse requirements of different classes. When the traffic load increases to high, as seen in Fig. 2.a, AIMTWIN prioritises HD video traffic, achieving higher throughput for this class while maintaining results comparable to conventional routing algorithms for other traffic types.

As the topology shifts to the medium-scale Sprint network topology, the advantages of AIMTWIN become increasingly apparent, particularly under medium to high traffic loads, as shown in Fig. 2.b. At low traffic loads, all routing schemes, including AIMTWIN, deliver maximum throughput for both HD and SD traffic classes, ensuring that QoS-based traffic flows meet the requirements specified in Table II. This can be attributed to the algorithms' ability to efficiently route incoming flows while rejecting those that could cause congestion. However, as the traffic load escalates to medium and high

TABLE III: Averaged Estimated PSNR and MOS

		MHA			WSP			SWP			MIRA			AIMTWIN		
		low	**med**	**high**	**low**	**med**	**high**	**low**	**med**	**high**	**low**	**med**	**high**	**low**	**med**	**high**
GetNet Network Topology																
HD	**PSNR**	60	55.9	30.3	60	55.4	30.7	60	48.9	30.7	60	55.4	31.1	60	54.51	56.03
	MOS	*Exc.*	*Exc.*	*Fair*	*Exc.*	*Exc.*	*Fair*	*Exc.*	*Exc.*	*Fair*	*Exc.*	*Exc.*	*Fair*	*Exc.*	*Exc.*	*Exc.*
SD	**PSNR**	60	53.9	46	60	50.5	54.9	58.4	48.6	44.7	60	52.8	58.4	58.78	57.26	48.04
	MOS	*Exc.*	*Exc.*	*Exc.*	*Exc.*	*Exc.*	*Exc.*	*Exc.*	*Exc.*	*Good*	*Exc.*	*Exc.*	*Exc.*	*Exc.*	*Exc.*	*Exc.*
Sprint Network Topology																
HD	**PSNR**	50.5	26.6	23.4	51.4	26.8	25.2	57.7	27.8	25	52	28.3	24.9	57.33	45.32	44.12
	MOS	*Exc.*	*Poor*	*Poor*	*Exc.*	*Poor*	*Poor*	*Exc.*	*Fair*	*Poor*	*Exc.*	*Fair*	*Poor*	*Exc.*	*Exc.*	*Good*
SD	**PSNR**	52.7	47.9	47.3	53.5	49.8	44	59.1	44	46.1	53.1	51.7	48.6	56.77	47.78	42.15
	MOS	*Exc.*	*Exc.*	*Exc.*	*Exc.*	*Exc.*	*Good*	*Exc.*	*Good*	*Exc.*	*Exc.*	*Exc.*	*Exc.*	*Exc.*	*Exc.*	*Good*

levels, the network experiences heightened resource competition among traffic classes. In these scenarios, AIMTWIN continues to prioritise HD video traffic, ensuring increased throughput, whereas conventional algorithms exhibit a decline in throughput for the HD traffic class.

Table III further highlights the estimated PSNR and MOS values when analyzing the classes of HD and SD video traffic in both network topologies. Even within the GetNet topology, AIMTWIN consistently achieves an *"Excellent"* user-perceived quality for both HD and SD video traffic across low, medium, and high traffic loads, without significantly penalising other traffic types. By contrast, as traffic load increases from medium to high, conventional routing algorithms such as MHA, WSP, SWP, and MIRA experience a degradation in user-perceived quality, with MOS scores dropping from *"Excellent"* to *"Fair"* for HD video, though SD traffic maintains an *"Excellent"* rating. The Sprint topology presents more pronounced differences in performance. While all routing algorithms achieve *"Excellent"* MOS ratings under low traffic loads, their performance diverges under medium and high traffic loads. AIMTWIN maintains a gradual decrease in MOS for HD traffic, transitioning from *"Excellent"* under low and medium loads to *"Good"* under high load. In contrast, conventional algorithms suffer a drastic decline, with MOS scores for HD traffic dropping from *"Excellent"* under low load to *"Poor"* under high load, underscoring their inability to manage congestion effectively.

Figure 3 evaluates the performance of the proposed AIMTWIN framework alongside other routing algorithms, focusing on rejection rates across various traffic classes, network topologies, and traffic loads. Figures 3.a and 3.b illustrate the total number of flows supported by the GetNet and Sprint topologies, respectively, as traffic load increases from low to high. Figures 3.c and 3.d depict the number of rejected flows for the GetNet and Sprint topologies, respectively, under the same escalating traffic conditions, detailing results for each traffic class and routing strategy. At high traffic loads, the rejection rates for all solutions rise significantly. Despite this, AIMTWIN effectively manages flow prioritisation, rejecting a similar number of HD video traffic flows compared to conventional algorithms while rejecting more non-critical background traffic. This behaviour underscores AIMTWIN's ability to outperform other state-of-the-art solutions by prioritising HD video flows without disproportionately compromising the performance of other traffic classes.

In comparison to AIMTWIN, conventional routing algorithms exhibit a tendency to accommodate more QoS-based flows at the expense of user-perceived QoE. This results in a significant degradation of MOS scores, as seen in the Sprint topology, where MOS for HD video traffic plummets from *"Excellent"* under low loads to *"Poor"* under high loads. Consequently, the conventional routing algorithms fail to meet QoS requirements for critical traffic classes under high load conditions, unlike AIMTWIN, which achieves a more robust and balanced performance across diverse scenarios.

VI. Conclusions and Future Work

This paper presents AIMTWIN, a novel RL-driven Digital Twin framework that addresses the challenges of multimedia traffic management in B5G SDN networks. By leveraging a tightly integrated feedback loop between the physical and virtual environments, AIMTWIN achieves dynamic and efficient routing of background traffic while maintaining static paths for QoS-sensitive flows. This tightly integrated loop between the Physical and Digital Twins ensures that AIMTWIN remains adaptive, efficient, and capable of managing the stringent requirements of multimedia traffic in B5G SDN networks. Experimental results validate the framework's ability to outperform conventional routing algorithms, ensuring high QoS, user satisfaction, and network resource utilisation. Future work will look at incorporating AIMTWIN within 6G environments to explore its adaptability to emerging use cases like holographic communications.

Acknowledgment

This research was carried out as part of the UKIERI-SPARC project 'DigIT' [Grant ID: UKIERI-SPARC/01/23] and partly supported by EPSRC UK-India Future Networks Initiative project 'Q-DTRAN' [Grant ID: EP/W016524/1].

References

[1] H. Adhami, M. Alja'afreh, M. Hoda, J. Zhao, Y. Zhou, and A. El Saddik, "Suitability of SDN and MEC to Facilitate Digital Twin Communication over LTE-A," *Digital Communications and Networks*, vol. 10, no. 2, pp. 347–354, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352864823000901

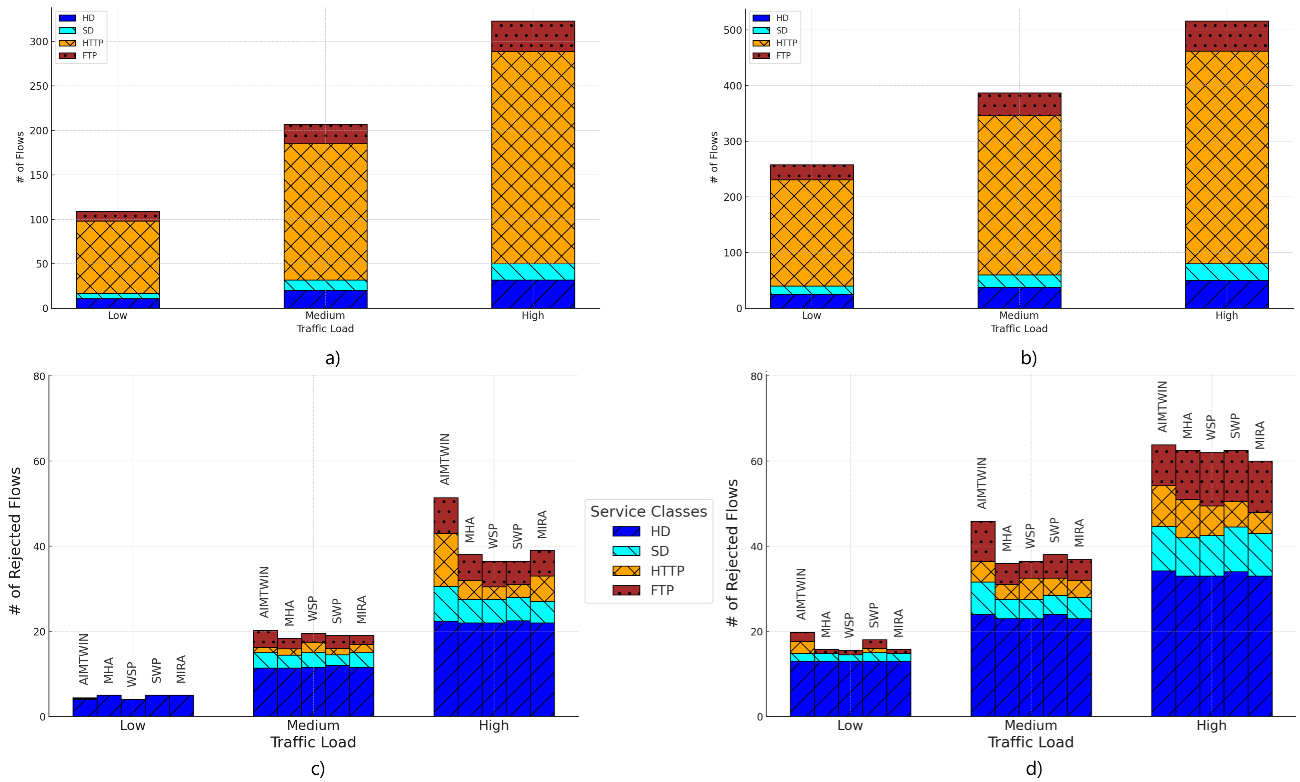

Fig. 3: The Total Number of Flows that are Generated in the Experiment Test for a) GetNet and b) Sprint Topologies. The total number of rejected flows in c) GetNet and d) Sprint Topologies.

[2] F. Naeem, G. Kaddoum, and M. Tariq, "Digital Twin-empowered Network Slicing in B5G Networks: Experience-driven Approach," in *IEEE Globecom Workshops (GC Wkshps)*, 2021, pp. 1–5.

[3] A. Karamchandani, M. Sanz, Burgaleta, L. De La Cal, A. Mozo, J. I. Moreno, A. Pastor, and D. R. López, "Design of an AI-Driven Network Digital Twin for Advanced 5G-6G Network Management," in *IEEE Network Operations and Management Symposium (NOMS)*, 2024, pp. 1–7.

[4] M. Sanz Rodrigo, D. Rivera, J. I. Moreno, M. Àlvarez Campana, and D. R. López, "Digital Twins for 5G Networks: A Modeling and Deployment Methodology," *IEEE Access*, vol. 11, pp. 38 112–38 126, 2023.

[5] R. Ramirez, C.-Y. Huang, and S.-H. Liang, "5G Digital Twin: A Study of Enabling Technologies," *Applied Sciences*, vol. 12, no. 15, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/15/7794

[6] L. Tang, Q. Hou, W. Wen, D. Fang, and Q. Chen, "Digital-Twin-Assisted VNF Migration Through Resource Prediction in SDN/NVF-Enabled IoT Networks," *IEEE Internet of Things Journal*, vol. 11, no. 21, pp. 35 445–35 464, 2024.

[7] H. X. Nguyen, R. Trestian, D. To, and M. Tatipamula, "Digital Twin for 5G and Beyond," *IEEE Communications Magazine*, vol. 59, no. 2, pp. 10–15, 2021.

[8] X. Lin, L. Kundu, C. Dick, E. Obiodu, T. Mostak, and M. Flaxman, "6G Digital Twin Networks: From Theory to Practice," *IEEE Communications Magazine*, vol. 61, no. 11, pp. 72–78, 2023.

[9] A. Mozo, A. Karamchandani, M. Sanz, J. I. Moreno, and A. Pastor, "B5GEMINI: Digital Twin Network for 5G and Beyond," in *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, 2022, pp. 1–6.

[10] L. U. Khan, W. Saad, D. Niyato, Z. Han, and C. S. Hong, "Digital-Twin-Enabled 6G: Vision, Architectural Trends, and Future Directions," *IEEE Communications Magazine*, vol. 60, no. 1, pp. 74–80, 2022.

[11] E. Boffetti, A. Rago, G. Piro, and G. Boggia, "Design of AI-based Digital Twin Network for Multimedia Service Provisioning," in *IEEE Symposium on Computers and Communications (ISCC)*, 2024, pp. 1–6.

[12] W. Qu, X. Li, and W. Chen, "A General Digital Twin Framework for Intelligent Network Management," in *IEEE/CIC International Conference on Communications in China (ICCC)*, 2022, pp. 760–765.

[13] I. S. Comşa, M. Aydin, S. Zhang, P. Kuonen, and J.-F. Wagen, "Reinforcement Learning Based Radio Resource Scheduling in LTE-Advanced," in *17th International Conference on Automation and Computing*, 2011.

[14] I. S. Comşa, M. Aydin, S. Zhang, P. Kuonen, J.-F. Wagen, and Y. Lu, "Scheduling Policies Based on Dynamic Throughput and Fairness Tradeoff Control in LTE-A Networks," in *39th Annual IEEE Conference on Local Computer Networks*, 2014, pp. 418–421.

[15] Z. Zhang, Y. Huang, C. Zhang, Q. Zheng, L. Yang, and X. You, "Digital Twin-Enhanced Deep Reinforcement Learning for Resource Management in Networks Slicing," *IEEE Transactions on Communications*, vol. 72, no. 10, pp. 6209–6224, 2024.

[16] L. Zhao, G. Han, Z. Li, and L. Shu, "Intelligent digital twin-based software-defined vehicular networks," *IEEE Network*, vol. 34, no. 5, pp. 178–184, 2020.

[17] I.-S. Comşa, S. Zhang, M. Aydin, P. Kuonen, R. Trestian, and G. Ghinea, "A Comparison of Reinforcement Learning Algorithms in Fairness-Oriented OFDMA Schedulers," *Information*, vol. 10, no. 10, 2019.

[18] I.-S. Comşa, S. Zhang, M. E. Aydin, P. Kuonen, Y. Lu, R. Trestian, and G. Ghinea, "Towards 5G: A Reinforcement Learning-Based Scheduling Solution for Data Traffic Management," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1661–1675, 2018.

[19] A. Al-Jawad, I.-S. Comşa, P. Shah, O. Gemikonakli, and R. Trestian, "REDO: A Reinforcement Learning-based Dynamic Routing Algorithm Selection Method for SDN," in *IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2021, pp. 54–59.

[20] A. Al Jawad, I.-S. Comşa, P. Shah, O. Gemikonakli, and R. Trestian, "An Innovative Reinforcement Learning-Based Framework for Quality of Service Provisioning Over Multimedia-Based SDN Environments," *IEEE Transactions on Broadcasting*, vol. 67, no. 4, pp. 851–867, 2021.

[21] A. Al-Jawad, I.-S. Comşa, P. Shah, and R. Trestian, "LearnSDN: Optimizing Routing Over Multimedia-based 5G-SDN using Machine Learning," in *14th International Conference on Communications (COMM)*, 2022, pp. 1–6.

[22] T. Zinner, O. Abboud, O. Hohlfeld, T. Hossfeld, and P. Tran-Gia, "Towards QoE Management for Scalable Video Streaming," in *21th ITC Specialist Seminar on Multimedia Applications-Traffic, Performance and QoE*, 2010, pp. 64 – 69.

[23] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The Internet Topology Zoo," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1765–1775, 2011.