**Swarm intelligence-based packet scheduling for future intelligent networks**

Arif Husen1,2, Muhammad Hasanain Chaudary1, Farooq Ahmed1, Muhammad Farooq-i-Azam3, Chan H. See4 and Arfan Ghani5

1 Dept. of Computer Science, COMSATS University Islamabad, Lahore Campus, Pakistan

2 Dept. of Computer Science and Information Technology, Virtual University of Pakistan.

3 Dept. of Electrical and Computer Engineering, COMSATS University Islamabad, Lahore Campus, Pakistan

4 School of Computing, Engineering and the Built Environment, Edinburgh Napier University, Edinburgh, Scotland, EH10 5DT, UK.

5 Dept. of Computer Science and Engineering, School of Engineering, American University of Ras al Khaimah, UAE

Corresponding Author:

Arif Husen1

Dept. of Computer Science, COMSATS University Islamabad, Lahore Campus, Pakistan.

Email address: arifhrashid@gmail.com

**Abstract**

Network operations involve several decision-making tasks. Some of these tasks are related to operators, such as extending the footprint or upgrading the network capacity. Other decision tasks are related to network functions, such as traffic classifications, scheduling, capacity, coverage trade-offs, and policy enforcement. These decisions are often decentralized, and each network node makes its own decisions based on the preconfigured rules or policies. To ensure effectiveness, it is essential that planning and functional decisions are in harmony. However, human intervention-based decisions are subject to high costs, delays, and mistakes. On the other hand, machine learning has been used in different fields of life to automate decision processes intelligently. Similarly, future intelligent networks are also expected to see intense use of machine learning and artificial intelligence techniques for functional and operational automation. This article investigates the current state-of-the-art methods for packet scheduling and related decision processes. Furthermore, it proposes a machine learning-based approach for packet scheduling for agile and cost-effective networks to address various issues and challenges. The analysis of the experimental results shows that the proposed deep learning-based approach can successfully address the challenges without compromising the network performance. For example, it has been seen that with mean absolute error from 6.38 to 8.41 using the proposed deep learning model, the packet scheduling can maintain 99.95% throughput, 99.97 % delay, and 99.94% jitter, which are much better as compared to the statically configured traffic profiles.

**Introduction**

Different network resources such as bandwidth, power, and spectrum are never abundant, regardless of the significant developments in the network technologies, to provide higher capacities and minimize delays (J. Huang et al., 2023; OECD, 2022; Shen & Chen, 2018; Sheng, Zhou, Liu, Wang, & Li, 2019). On the other hand, the traffic growth is immense due to rising trends in digitization and network-enabled devices from the Internet of Vehicles (IoV), Tactile Internet (TI), and Internet of Things (IoT) (Cisco, 2021). Moreover, not only the new application ‘s underlying reasons but also the bandwidth requirements from the common services are rising due to increased quality and data and latency constraints. As a result, network operators are often challenged to implement suitable oversubscription models (Ni, Huang, & Wu, 2014) and differentiated service bundling mechanisms that allow them to ensure Return on Investment (ROI). For example, packet scheduling (Beams, Kannan, & Angel, 2021) is a common approach to controlling service quality over networks in oversubscribed networks where the traffic of different types needs to be handled according to their service package or the contracted Service Level Agreements (SLA).

Network deployments often follow a comprehensive planning process that involves several demographic and statistical factors in addition to seasonal and regular network user behaviors. Planning processes often utilize the type of services, the technologies, the required volume of network resources, and the target SLA (Gavriluţ, Pruski, & Berger, 2022). Traditionally for cellular networks, the erlang traffic intensities describe the expected load that networks must withstand in different types of Points of Presence (POP)(Głąbowski, Hanczewski, & Stasiak, 2015). The loads correspond to business models, target ROI, and other policies of the network operators.

Traditional packet scheduling techniques utilize several factors, including non-differentiated approaches where traffic scheduling takes place without differences, such as First-in-First-out (FIFO), Fair Queueing (FQ), and Priority Queuing (PQ), and Round Robin (RR)(Deep, 2018). Differentiated approaches include classifying traffic into different classes and assigning a weight or priority to the classes. The later approaches often employ the non-differentiated approaches within a class or priority group. Recently it has been studied that significant Quality of Service (QoS) and network utilization improvements can be made with packet scheduling approaches that consider the features such as Time and Origin Characteristics (TAOC) of Traffic (A. Husen et al., 2021; Rashid & Muhammad, 2019). TAOC characterization is related to the network planning processes, such as traffic forecasting and its breakdown to the level of traffic origins, formally known as POPs. The TAOC characterization concentrates on novel features such as Origin Class Feature (OCF), Volume Feature (VF), Time Feature (TF), Traffic Intensity Feature (TIF), and Network Resource Feature (NRF).

This article investigates the Deep Learning (DL) based techniques for learning the TAOC-based packet scheduling algorithms. Following are the contributions of this research,

1. The existing packet scheduling approaches are analyzed in the context of intelligent methods.
2. A novel DL-based traffic profiling scheme is proposed to fulfill the swarm intelligence requirement of traffic intensity-based packet scheduling.
3. The traffic intensity-based packet scheduling algorithm is ported to Network Simulator 3 (NS3) for experimental evaluations and analysis.
4. An analysis of the DL-based profiling scheme and network performance objectives is presented.

Contributions (1) and (2) are required for implementing TAOC-aware packet scheduling to bridge the gap between network planning processes and functional network decision-making, such as packet scheduling. The framework also considers network node role awareness while making packet scheduling decisions.

The rest of the article is divided into the materials and methods, results, discussion, and conclusion. First, the material and methods section covers the packet scheduling strategies, learning mechanism for TAOC, high-level framework for TAOC-based packet scheduling, Deep Learning (DL) based model, and its integration with the TAOC-based packet scheduling algorithm. Next, the results section presents the experimental setup and evaluation metrics for the DL model and network performance. Finally, the discussion sections focus on the significance of the results and future research directions.

**Materials & Methods**

**I - Packet** **Scheduling** **Approaches**

There is a large number of packet scheduling techniques that work based on traditional factors such as weight, priority, class, and arrival basis. However, there are very few techniques proposed that consider TAOC characteristics. The prominent scheme in this regard was proposed by (A. Husen et al., 2021; Rashid & Muhammad, 2019) that considers the TAOC characteristics and network node and layer role; however, the study did not consider the effects of any machine learning-based real-time TAOC characterization scheme. Therefore, a comparison of existing techniques is given in **Table 1**.

The traditional packet scheduling schemes can be categorized into reactive and provocative. The reactive schemes monitor certain parameters, and on change, the scheduling decisions are updated, such as in Relative Differentiated Scheduling (RDS) (Striegel & Manimaran, 2002), Preemption-based scheduling (PBS),(Miao, Min, Wu, & Wang, 2015), and Queue Length (QL)-based Delay-aware Packet Scheduler (QLDA)(Q. Yu, Znati, & Yang, 2015). In addition, the Active Time Fair Queuing (ATFQ) (J. Zhang et al., 2015), Competitive Rate Based Scheduling (CRBS) (Deshmukh & Vaze, 2016), DWLC-FQ(Patel & Dalal, 2016), TFRC(Sungjoo, Kim, Chang Mo, & Chai-Jong, 2016), Modified FCFS,(Xu, Chang, Lin, Shen, & Zhu, 2016) and Eiffel are also reactive schemes proposed in the recent literature(Vahdat, 2019). On the other hand, the proactive schemes adjust the decision parameters before the changes occur, such as in the multiple dimensions of locality-based scheduling(Iqbal, Holt, Ryoo, Veciana, & John, 2016), BSA, Multi-Generation packet scheduling(S. Huang, Izquierdo, & Hao, 2017), D2-Pas(T. Zhang, Gong, Han, Deng, & Hu, 2019), Calendar Queing(Naveen Kr. Sharma; Chenxingyu Zhao; Ming Liu; Pravein G. Kannan; Changhoon Kim; Arvind Krishnamurthy, 2020) and TIPS (A. Husen et al., 2021; Rashid & Muhammad, 2019).

It can be observed from the above analysis that few techniques consider the TAOC features, which are partial except for the TIPS schemes. The TIPS technique was implemented and evaluated with predefined TAOC features and did not learn from actual live streams of the traffic. The predefined TAOC features are a tedious and complex process, and it may take extensive time to conclude the features and require domain knowledge and human intervention.

**II - Learning** **TAOC** **Features**

Learning the TAOC characteristics involves classifying traffic according to its volumes, origin, and periodic variations as shown in **Figure 1**. In addition, suitable techniques are required to approximate the historical TOAC features and use the features to predict the traffic intensity at a given or future point in time.

In the recent era, Future Intelligent Networks (FIN) (Arif Husen, Chaudary, & Ahmad, 2022) have been envisioned, which are expected to use ML techniques to automate computation, network functions, and operations intelligently. For example, the ML-based intelligent packet scheduling techniques shall learn the network state and usage behavior and control the scheduling decisions on the different types of network nodes. ML network techniques provide autonomous, cheaper, and faster decision-making in managing network functionalities.

Recent literature has several studies on ML-based techniques for traffic classification and forecasting. The ML models used in the above studies classify the traffic streams. In addition, they forecast their behavior in the future time based on learning the insights from historical usage of the data. The ML models have been used to extract the Spatiotemporal features, some of which are also parts of TAOC characterization; however, they have not yet been considered for packet scheduling decisions. For TAOC, specific ML models are required for traffic classification based on their origin. This classification differs from the traditional classification used for packet scheduling, which is flow-based or source/destination addresses, port numbers, or applications. For origin-based traffic classification, the classes represent the different groups based on the origin areas, such as serving areas or POPs. The time feature represents the traffic variations per origin class (OCF) and captures how traffic patterns change with time, day, week, month, year, or season. The Volume Feature (VF) captures the variations of the traffic volumes in terms of the number of packets. It also needs to cover the packet size variations, thus covering the maximum capacity the switching system has to handle.

Traffic Intensity Features involve deriving the Erlang values for the given OC and VF. Above mentioned features are used for OC trend curves, and packet schedulers generate the queues according to the number of Ocs. Packets are processed from the input interfaces to output interfaces in precedence of the predicted values. It can be noticed in **Table 2** that existing works for traffic classification and prediction do not cover all the features required for TAOC characterization. However, the studies consider several partial requirements, such as Spatio-temporal features addressed by Zhang et al.(C. Zhang, Zhang, Qiao, Yuan, & Zhang, 2019), He, Chow & Zhang (He, Chow, & Zhang, 2019), Bega et al.(Bega, Gramaglia, Fiore, Banchs, & Costa-Perez, 2019), Wang et al. (J. Wang et al., 2017) and Daniel (Wass, 2021). However, these studies do not cover the whole scope of traffic forecasting. For example, they cannot generate the classes of traffic based on OC and their approximations using Time, Volume, and Traffic Intensity.

Similar is the case with traffic classification, addressed in several existing studies such as (Nguyen, Armitage, Branch, & Zander, 2012), (Singh & Agrawal, 2011), (C. Zhang et al., 2019), and (Reddy & Hota, 2013). However, the scope of these studies is very limited to identifying one class vs. other specific application traffic. Therefore, we reiterate that further studies are required to investigate traffic classification and forecasting based on the TAOC features, which differ from the application or type of traffic classification and prediction.

**III - High-Level Framework for TAOC-Based Packet Scheduling**

This section focuses on a high-level framework for ML-based TAOC packet scheduling, which covers the serval outstanding aspects of packet scheduling. The benefits of the TAOC features include user traffic classification, prediction, and network state regarding bandwidth capacities. The high-level implementation of the ML-based TAOC Packet Scheduling is shown in **Figure 2**. It employs several network layer-specific learners, and the learning process and forecasting process if distributed.

The base learns (BL) provide the OC classification and approximation of the TF, VF, and TIF features. The predictions from BLs are used to manipulate scheduling on perimeters of networks, such as the access or backhaul nodes.

Distribution-aggregate learners (DAL) aggregate the learnings from the BLs. The DAL does not learn from the actual traffic on distribution or aggregation nodes; rather, it employs the learnings from the BLs and provides the necessary information for deciding on distribution nodes. The DALs are node role-specific learners and incorporate the effects of oversubscription.

Similarly, the Core-Aggregate Leaner (CAL) and Edge-Aggregate Learners (EAL) aggregate the learning from the DAL and incorporate the node role and oversubscription effects. The EALs are based on CAL learnings and incorporate the breakout ratio of the traffic, i.e., the traffic that will leave the network perimeters.

The predictions from each of the learners are inputted to the respective TIPS schedulers (Rashid & Muhammad, 2019), such as Access-Packet Schedular (APS), Distribution-Packet Schedular (DPS), Core-Packet Schedular (CPS), Edge-Packet Schedulers (EPS).

**IV - TAOC Learning with LSTM**

Erlang distribution incorporates the TAOC characteristics and Recurrent Neural Networks (RNN)(Jain & Medsker, 1999), as shown in **Figure 3** deep learning paradigm suitable for predicting the Erlang distribution based on the traffic characteristics of each access node. Moreover, since the traffic originating from a given node depends on the previous pattern, the patterns are generally interlinked and may also repeat in future time intervals. Finally, the traffic patterns depend on the serving area's demographics. Traditional neural networks can be used to characterize the traffic with temporal dependencies; however, RNNs, due to their specific architecture, can address the issues.

**(a) - An Analysis of LSTM**

Long Short-Term Memory (LSTM)(Houdt, Mosquera, & Nápoles, 2020) is a special RNN cell that addresses the issues faced with traditional RNNs, such as loss of the long-term dependencies due to vanishing gradient issues, as discussed in (Bengio, Simard, & Frasconi, 1994).

**Figure 4** shows that an LSTM employs three gates to control adding or removing information from the cell state. The forget gate () defines what information needs to be removed from the cell state. It analyses the previous state () and current input sequence ().

The is defined in Equation (1), and its output is always between 0 and 1 for each state in the where zero means discarding it.

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

The information that needs to be retained in the current state of the cell is determined in two steps. First, the input gate () function, as defined in Equation (2), determines the parameters that need to be updated, and candidate parameter values () are generated by the tanh function as per Equation (3) which are combined to update the values of as per Equation (4).

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

The output of the LSTM is based on the current state, and the sigmoid function is applied to select the values to be output. The final output of the cell is obtained by applying the tanh function to the above values and multiplied by the output gate. The output gate function is defined in Equation (5), and the current output state is defined in Equation (6).

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

**(b) - TAOC Learning Model (TLM)**

TIPS scheduling algorithm requires five different values, including the time, node function, number of downstream nodes, and rank and weight of the Erlang values. Therefore, the objective of the TLM is to characterize each access node and provide above mentioned information to the upstream node. To cope with the above requirements, an LSTM-based encoder-decoder model is adopted, as shown in **Figure 5**.

The above model is replicated for each physical interface on the downstream side. In addition to the unit model for each interface, an aggregate model combines the input/output values. The model takes the number of sample values of and predicts the number of values for .

In the unit model, represents the LSTM cells for the encoder where ranges in and represents the LSTM cells for the decoder part. The ranges from to . The forms the dense layer, which receives the state information from all the cells and transforms it into the output of Dimension. On the other hand, is a dense layer (Helen Josephine, Nirmala, & Alluri, 2021) that receives the m inputs and connects the F layer to cconnecta single dimension stream. The flattening layer (Chen et al., 2023) that receives the inputs equals the hidden layers () multiplied by the units.

The TLM is installed on each real network node with the downstream nodes, such as the aggregation layer (DAL) and core nodes (CAL). The core nodes receive the swarm of g inputs fed to TLM(c) nodes. The TLM(g) nodes provide the data to TIPS instances running on G nodes, and TLM(c) provides the data for the TIPS instances running on C nodes, as shown in **Figure 6**.

There are two approaches available for implementing TLM on the DAL layer. The first approach is to install TLM on the egress interface of the access nodes, which predicts the used by the TLM instances running on DAL. This approach is efficient for large-scale networks and allows the distributed learning model. The other approach is to implement TLM on the ingress interfaces of aggregation nodes, i.e., DAL nodes. In this case, it serves two purposes; the TLM provides the base learning as well as in addition to .

**Results**

This section covers the details of the experimental setup, various parameters, evaluation metrics, and their measured results. First, Mean Absolute Error (MAE) and Mean Squared Error (MSE) are used for the TLM evaluation to determine the difference from the actual traffic features. Next, the throughput, delay, and jitter values are measured for the network performance. Finally, the Cumulative Distribution Function (CDF)(Tuma, 2005) is used to depict the results of different measurements, as it can show how the different metrics vary throughout the experiment.

**I - Experimental Setup**

The TLM was implemented using the KERAS library (Gulli, 2017) in Python for experimental analysis, and the source code is available in annexure (B). The TIPS was originally implemented for NS2 and was ported to Network Simulator 3 (NS3) (Riley & Henderson, 2010) to integrate the TLM.

***(a) - Network Topology***

**Figure 7** shows the experimental network setup following the state-of-the-art hierarchical topology. The hierarchal network topologies offer several benefits in scalability, redundancy, performance, security, manageability, and maintenance (Cisco, 2022; Jinxin Zhang & Liang, 2008). The access nodes represent a serving area with a distinct traffic profile. The access nodes to aggregation nodes connectivity are with dedicated physical links with different data rates according to the requirements of the serving area. Furthermore, the access node to aggregation nodes connectivity follows the rule of proximity, i.e., the access nodes from colocating areas are connected to the same aggregation node.

For the experimental setup, traffic sources are configured to generate the traffic with Gaussian distribution, a well-known internet traffic distribution discussed in (Bothe, Qureshi, & Imran, 2019; Meent, Mandjes, & Pras, 2006). Although all the access nodes will follow a Gaussian distribution, their mean and shape are different for different types of demographics of the serving area, as discussed in a study conducted by (A. Husen et al., 2021).

The specifications of different links are given in **Table 3**. The access to aggregation links follows random allocation for each experiment instance and uses a FIFO scheduler. The TIPS is not used on access links as it represents a single traffic profile. On aggregation to core nodes, the link's capacity is set to 4Mbps aggregation to core links has 6Mbps capacity. Both of the above links use the TIPS packet scheduler. The edge links connecting the destination nodes use the FIFO schedular as there is only a single type of traffic profile.

***(b) – Traffic Intensity Based Packet Scheduling (TIPS)***

The TIPS is a packet schedular that schedules the packet according to the traffic profile of downstream links. Packets are enqueued to different queues with dynamic dequeue order following the ranks of the traffic profile, and the size of the dequeue size is dependent on the weights of traffic profiles. The control traffic is separated and handled with different policies different from TIPS. The TIPS was initially developed for NS2 (A. Husen et al., 2021); however, since the NS2 development has stopped, it was ported to Network Simulator 3 (NS3) (Riley & Henderson, 2010). The implementation of TIPS for NS3 is available in the supplementary material of this paper.

**II - TLM Performance Results and Analysis**

This section focuses on analyzing the traffic generated from the serving areas, evaluating the performance of TLM on aggregation and core nodes, and finally, the network performance analysis, which includes throughput, delay, and jitter. Finally, the network performance shall be compared with results obtained without TLM.

**(a) - Access Traffic Profiles**

The TLM performance's primary expectation is traffic profile diversity. The traffic profiles used in the experiments are shown in **Figures 8 to 10** for the difference aggregation nodes, as labeled below. The graphs in the above figure show the cumulative density of the data rates generated by the respective node along the histogram.

The histogram on the top part of the graph shows that all access nodes follow a Gaussian distribution (Yamanaka & Usuba, 2020) which means that for the given mean, their equal chances of traffic generated below the mean or above the mean.

**(b) - Prediction Accuracy of TLMs**

This section analyses the performance of the TLM models on the different nodes. The TLM learners are used on G1, G2, G3, and C1 as per the selected links. The prediction accuracy of different TLMs is shown in **Figures 11 to 14**. The figures present the histogram of the actual data rates and those predicted by the respected TLM. A summary of the mean absolute difference (MAE) and mean squared error (MSE) difference between the actual data rate is given in **Table 4** which shows the MAE values between 8.41 to 6.38 for TLM on nodes G1, G2, G3, and C1. Since the TLMs coordinate with each other to realize swarm learning, the latency of the inter TLMs may pose a limitation. In practice to address it, low latency dedicated communication links can be used. The latency issues between inter-TLM communications can also be minimized by implementing the TLMs on the same nodes or the same locations as the nodes of the network. The location of TLMs for different network nodes is an essential factor to ensure optimal performance. In this study, the TLMs were implemented in the same location of the respective network nodes to eliminate the effects of the latency. If the TLMs are implemented in a centralized location to avail benefits of the cloud computing paradigm, low latency dedicated networks would be required.

**III - Network Performance Analysis of TLM**

This section evaluates the network performance with TIPS with TLM as a packet scheduler on links on aggregation and core nodes. The throughput, source-to-destination delay, and jitter are measured with and without TLM.

**Figure 15** shows the maximum throughput achieved with TLM and without TLM. Without TLM, TIPS uses statically configured profiles. The TLM learns the traffic profiles of access nodes on run time and provides necessary information for scheduling decisions. It can be seen that the TLM achieves the same throughput as the manually engineered profiles. For both cases, the throughput achieved was 2000 Kbps for almost 50% of measurements. The average difference with the TLM algorithm is observed to be 0.05%.

Similarly, **Figure 16** shows the maximum delay achieved with and without TLM. Without TLM, TIPS uses statically configured profiles. It can be seen that the TLM maintains the same delay as the manually engineered profiles. For both cases, the maximum delay is centered around 15 to 17.5ms. The average delay difference with TLM is observed to be 0.03%.

Finally, **Figure 17** shows the maximum jitter achieved with and without TLM. Without TLM, TIPS uses statically configured profiles. Similarly to delay results, It can be seen that the TLM maintains the same jitter as the manually engineered profiles. The maximum jitter is around 2ms for both of the cases. The average jitter difference with TLM is 0.05% which shows that it can maintain jitter up to 99.95%.

**Discussion**

The TIPS was compared with several state-of-art scheduling algorithms (Husen, A. et al., 2021) and showed that TIPS provides the advantage in terms of throughput, delay, and jitter in congested networks. This research has evaluated the automated traffic profile learning with the Deep Learning Model, TLM, and compared it with traffic profiles statically configured on the network nodes through the network dimensioning processes. The efficiency of TLM in learning the TAOC characteristics is demonstrated and shows that performance is better than the statically configured profiles.

The intelligent automation of the various future network functions and processes is an essential requirement. It has been envisioned in several recent articles such as (Brito, Mendes, & Gontijo, 2020; S. Wang, Sun, Yang, Duan, & Lu, 2020; Zhu, Zhao, Zhang, & Zhou, 2020), where ML-based automation has been indicated as the primary requirement. Thus avoiding manually configured traffic profiles is a significant challenge in future intelligent networks especially the human-based network dimensioning processes are expensive in terms of cost, time, and agility. The TLM can intelligently automate the traffic profile learning process with distributed learning architecture, which eliminates human intervention and makes the network agile to user behavior and network changes.

The results presented in the previous subsections show that the same performance can be obtained with TLM, thus eliminating the costly processes of traffic engineering or profile building. The aspects of TLM coordination and related limitations such as the latency and detection of traffic profiles have also been evaluated. Several experiments were conducted to show the effectiveness of the automated approach with TLM. Moreover, it saves costs, automates traffic profile learning, and makes networks responsive to network dimensionality factors, such as a change in traffic.

The TLM model evaluated in this research is an RNN-based model that is well known to detect the temporal dependencies, however, the spatial dependencies in the traffic profiles may also exist. Future works in this direction include the incorporation of spatial learning models along temporal models such as convolutional neural networks. Since the conventional DL models lack the explainability of decisions predicted (Jialai W. et al. 2022), the objective function-based feature engineering to make explainable decisions requires further studies.

**Conclusions**

Packet scheduling is an active area of research for mobile and fixed networks, and the researchers strive to improve the performance, QoS, and Network Utilization. TAOC characterization can overcome the shortcomings of traditional decentralized and independent techniques on each network node. Research has recently concentrated on such features to develop a swarm intelligence-based system. The TAOC characterization is an important area, and the manual procedures used in previous works are inefficient in traffic dynamics. The value of ML-based TAOC packet scheduling lies in its capability to bridge the gaps between network planning and functional decisions. The ML-based swarm intelligent packet scheduling framework, i.e., TLM and TIPS introduced in the article, can intelligently automate the TAOC characterization process and capture the traffic, network, and user dynamics in real time. To verify this, the TIPS was ported to NS3 for experimental purposes. The experiments' results have concluded that the proposed approach can address the challenges without affecting the network performance metrics. It has been seen that with MAE from 6.38 to 8.41 (both DAL and CAL) with the proposed TLM, the TAOC-based packet schedular can maintain throughput, delay, and jitter with less than 0.05% variation as compared to the statically configured traffic profiles.

**References**

Beams, A., Kannan, S., & Angel, S. (2021). Packet Scheduling with Optional Client Privacy. Paper presented at the Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea. https://doi.org/10.1145/3460120.3485371

Bega, D., Gramaglia, M., Fiore, M., Banchs, A., & Costa-Perez, X. (2019, 29 April-2 May 2019). DeepCog: Cognitive Network Management in Sliced 5G Networks with Deep Learning. Paper presented at the IEEE INFOCOM 2019 - IEEE Conference on Computer Communications.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2), 157-166. doi:10.1109/72.279181

Bothe, S., Qureshi, H. N., & Imran, A. (2019). Which Statistical Distribution Best Characterizes Modern Cellular Traffic and What Factors Could Predict Its Spatiotemporal Variability? IEEE Communications Letters, 23(5), 810-813. doi:10.1109/LCOMM.2019.2908370

Brito, J. M. C., Mendes, L. L., & Gontijo, J. G. S. (2020, 17-20 March 2020). Brazil 6G Project - An Approach to Build a National-wise Framework for 6G Networks. Paper presented at the 2020 2nd 6G Wireless Summit (6G SUMMIT).

Chen, H., Zhou, H., Zhang, J., Chen, D., Zhang, W., Chen, K., . . . Yu, N. (2023). Perceptual Hashing of Deep Convolutional Neural Networks for Model Copy Detection. ACM Trans. Multimedia Comput. Commun. Appl., 19(3), Article 123. doi:10.1145/3572777

Cisco. (2021). Global - 2021 Forecast Highlights. Retrieved from https://www.cisco.com/c/dam/m/en\_us/solutions/service-provider/vni-forecast-highlights/pdf/Global\_2021\_Forecast\_Highlights.pdf

Cisco. (2022). Benefits of a Hierarchical Network. Retrieved from http://cisco-training-academy.blogspot.com/2009/10/benefits-of-hierarchical-network.html

Deep, M. a. K., Ramasamy. (2018). Network Routing, Algorithms, Protocols, and Architectures: Elsevier

Deshmukh, A., & Vaze, R. (2016). Online Energy-Efficient Packet Scheduling for a Common Deadline With and Without Energy Harvesting. IEEE Journal on Selected Areas in Communications, 34(12), 3661-3674. doi:10.1109/JSAC.2016.2611899

Gavriluţ, V., Pruski, A., & Berger, M. S. (2022). Constructive or Optimized: An Overview of Strategies to Design Networks for Time-Critical Applications. ACM Comput. Surv., 55(3), Article 62. doi:10.1145/3501294

Głąbowski, M., Hanczewski, S., & Stasiak, M. (2015). Modelling of Cellular Networks with Traffic Overflow. Mathematical Problems in Engineering, 2015, 286490. doi:10.1155/2015/286490

Gulli, A. a. P., Sujit. (2017). Deep learning with Keras: Packt Publishing Ltd.

Han, D., You, K., Xie, L., Wu, J., & Shi, L. (2015, 15-18 Dec. 2015). Stochastic packet scheduling for optimal parameter estimation. Paper presented at the 2015 54th IEEE Conference on Decision and Control (CDC).

He, Z., Chow, C. Y., & Zhang, J. D. (2019, 10-13 June 2019). STCNN: A Spatio-Temporal Convolutional Neural Network for Long-Term Traffic Prediction. Paper presented at the 2019 20th IEEE International Conference on Mobile Data Management (MDM).

Helen Josephine, V. L., Nirmala, A. P., & Alluri, V. L. (2021). Impact of Hidden Dense Layers in Convolutional Neural Network to enhance Performance of Classification Model. IOP Conference Series: Materials Science and Engineering, 1131(1), 012007. doi:10.1088/1757-899X/1131/1/012007

Houdt, G. V., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. Artif. Intell. Rev., 53(8), 5929–5955. doi:10.1007/s10462-020-09838-1

Huang, J., Yang, F., Chakraborty, C., Guo, Z., Zhang, H., Zhen, L., & Yu, K. (2023). Opportunistic capacity based resource allocation for 6G wireless systems with network slicing. Future Generation Computer Systems, 140, 390-401. doi:https://doi.org/10.1016/j.future.2022.10.032

Huang, S., Izquierdo, E., & Hao, P. (2017). Adaptive packet scheduling for scalable video streaming with network coding. Journal of Visual Communication and Image Representation, 43, 10-20. doi:https://doi.org/10.1016/j.jvcir.2016.11.014

Husen, A., Chaudary, M. H., & Ahmad, F. (2022). A Survey on Requirements of Future Intelligent Networks: Solutions and Future Research Directions. ACM Comput. Surv. doi:10.1145/3524106

Husen, A., Hasanain Chaudary, M., Ahmad, F., Imtiaz Alam, M., Sohail, A., & Asif, M. (2021). Improving scheduling performance in congested networks. PeerJ Comput Sci, 7, e754. doi:10.7717/peerj-cs.754

Iqbal, M. F., Holt, J., Ryoo, J. H., Veciana, G. d., & John, L. K. (2016). Dynamic Core Allocation and Packet Scheduling in Multicore Network Processors. IEEE Transactions on Computers, 65(12), 3646-3660. doi:10.1109/TC.2016.2560838

Jain, L. C., & Medsker, L. R. (1999). Recurrent Neural Networks: Design and Applications: CRC Press, Inc.

Karimi, A., Pedersen, K. I., Mahmood, N. H., Pocovi, G., & Mogensen, P. (2019, 28 April-1 May 2019). Efficient Low Complexity Packet Scheduling Algorithm for Mixed URLLC and eMBB Traffic in 5G. Paper presented at the 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring).

Lee, S. J., & Choi, Y. S. (2015, 28-30 Oct. 2015). GMPS(Group based multi-level packet scheduling) method in multi-beam based mobile communication system. Paper presented at the 2015 International Conference on Information and Communication Technology Convergence (ICTC).

Meent, R. V. D., Mandjes, M., & Pras, A. (2006, 11-15 June 2006). Gaussian traffic everywhere? Paper presented at the 2006 IEEE International Conference on Communications.

Miao, W., Min, G., Wu, Y., & Wang, H. (2015, 19-21 Dec. 2015). Performance Modelling of Preemption-Based Packet Scheduling for Data Plane in Software Defined Networks. Paper presented at the 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity).

Naveen Kr. Sharma; Chenxingyu Zhao; Ming Liu; Pravein G. Kannan; Changhoon Kim; Arvind Krishnamurthy, a. A. S. (2020). Programmable calendar queues for high-speed packet scheduling. Paper presented at the 17th USENIX Symposium on Networked Systems Design and Implementation. https://www.usenix.org/conference/nsdi20/presentation/sharma

Nguyen, T. T. T., Armitage, G., Branch, P., & Zander, S. (2012). Timely and Continuous Machine-Learning-Based Classification for Interactive IP Traffic. IEEE/ACM Transactions on Networking, 20(6), 1880-1894. doi:10.1109/TNET.2012.2187305

Ni, W., Huang, C., & Wu, J. (2014). Provisioning high-availability datacenter networks for full bandwidth communication. Computer Networks, 68, 71-94. doi:https://doi.org/10.1016/j.comnet.2013.12.006

OECD. (2022). The operators and their future: The state of play and emerging business models (20716826). Retrieved from https://www.oecd-ilibrary.org/science-and-technology/the-operators-and-their-future\_60c93aa7-en

Patel, Z., & Dalal, U. (2016). Implementation and evaluation of dynamically weighted low complexity fair queuing (DWLC-FQ) algorithm for packet scheduling in WiMAX networks. China Communications, 13(5), 128-140. doi:10.1109/CC.2016.7489981

Pavithira, S. D., & Prabakaran, N. (2016, 18-19 March 2016). Downlink packet scheduling mechanism in long term evolution technology. Paper presented at the 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT).

Rashid, A. H., & Muhammad, S. S. (2019, 20-21 March 2019). Traffic Intensity Based Efficient Packet Schedualing. Paper presented at the 2019 International Conference on Communication Technologies (ComTech).

Reddy, J. M., & Hota, C. (2013). P2P traffic classification using ensemble learning. Paper presented at the Proceedings of the 5th IBM Collaborative Academia Research Exchange Workshop, New Delhi, India. https://doi.org/10.1145/2528228.2528243

https://dl.acm.org/doi/pdf/10.1145/2528228.2528243

Riley, G. F., & Henderson, T. R. (2010). The ns-3 Network Simulator. In K. Wehrle, M. Güneş, & J. Gross (Eds.), Modeling and Tools for Network Simulation (pp. 15-34). Berlin, Heidelberg: Springer Berlin Heidelberg.

Shen, H., & Chen, L. (2018). Resource Demand Misalignment: An Important Factor to Consider for Reducing Resource Over-Provisioning in Cloud Datacenters. IEEE/ACM Transactions on Networking, 26(3), 1207-1221. doi:10.1109/TNET.2018.2823642

Sheng, M., Zhou, D., Liu, R., Wang, Y., & Li, J. (2019). Resource Mobility in Space Information Networks: Opportunities, Challenges, and Approaches. IEEE Network, 33(1), 128-135. doi:10.1109/MNET.2018.1700244

Singh, K., & Agrawal, S. (2011). Feature extraction based IP traffic classification using machine learning. Paper presented at the Proceedings of the International Conference on Advances in Computing and Artificial Intelligence, Rajpura/Punjab, India. https://doi.org/10.1145/2007052.2007095

https://dl.acm.org/doi/pdf/10.1145/2007052.2007095

Striegel, A., & Manimaran, G. (2002). Packet scheduling with delay and loss differentiation. Computer Communications, 25(1), 21-31. doi:https://doi.org/10.1016/S0140-3664(01)00337-1

Sungjoo, P., Kim, D., Chang Mo, Y., & Chai-Jong, S. (2016, 5-8 July 2016). A packet scheduling scheme for seamless transmission of life media contents. Paper presented at the 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN).

Tuma, N. B. (2005). Event History Analysis. In K. Kempf-Leonard (Ed.), Encyclopedia of Social Measurement (pp. 859-869). New York: Elsevier.

Vahdat, A. S. a. Y. Z. N. D. E. Z. a. M. A. K. H. A. (2019). Eiffel: Efficient and Flexible Software Packet Scheduling. Retrieved from https://www.usenix.org/conference/nsdi19/presentation/saeed

Wang, J., Tang, J., Xu, Z., Wang, Y., Xue, G., Zhang, X., & Yang, D. (2017, 1-4 May 2017). Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach. Paper presented at the IEEE INFOCOM 2017 - IEEE Conference on Computer Communications.

Wang, S., Sun, T., Yang, H., Duan, X., & Lu, L. (2020, 17-20 March 2020). 6G Network: Towards a Distributed and Autonomous System. Paper presented at the 2020 2nd 6G Wireless Summit (6G SUMMIT).

Wass, D. (2021). Transformer learning for traffic prediction in mobile networks. KTH Royal Institute of Technology, https://www.diva-portal.org/smash/get/diva2:1609275/FULLTEXT01.pdf. Retrieved from https://www.diva-portal.org/smash/get/diva2:1609275/FULLTEXT01.pdf

Wei, W., Xue, K., Han, J., Wei, D. S. L., & Hong, P. (2020). Shared Bottleneck-Based Congestion Control and Packet Scheduling for Multipath TCP. IEEE/ACM Transactions on Networking, 28(2), 653-666. doi:10.1109/TNET.2020.2970032

Xu, S., Chang, T. H., Lin, S. C., Shen, C., & Zhu, G. (2016). Energy-Efficient Packet Scheduling With Finite Blocklength Codes: Convexity Analysis and Efficient Algorithms. IEEE Transactions on Wireless Communications, 15(8), 5527-5540. doi:10.1109/TWC.2016.2561273

Yamanaka, S., & Usuba, H. (2020). Rethinking the Dual Gaussian Distribution Model for Predicting Touch Accuracy in On-screen-start Pointing Tasks. Proc. ACM Hum.-Comput. Interact., 4(ISS), Article 205. doi:10.1145/3427333

Yu, Q., Znati, T., & Yang, W. (2015, 14-16 Dec. 2015). Energy-efficient, Delay-aware packet scheduling in high-speed networks. Paper presented at the 2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC).

Yu, Z., Hu, C., Wu, J., Sun, X., Braverman, V., Chowdhury, M., . . . Jin, X. (2021). Programmable packet scheduling with a single queue. Paper presented at the Proceedings of the 2021 ACM SIGCOMM 2021 Conference, Virtual Event, USA. https://doi.org/10.1145/3452296.3472887

https://dl.acm.org/doi/pdf/10.1145/3452296.3472887

Zhang, C., Zhang, H., Qiao, J., Yuan, D., & Zhang, M. (2019). Deep Transfer Learning for Intelligent Cellular Traffic Prediction Based on Cross-Domain Big Data. IEEE Journal on Selected Areas in Communications, 37(6), 1389-1401. doi:10.1109/JSAC.2019.2904363

Zhang, J., & Liang, M. (2008). A hierarchical networking architecture based on new switching address. Paper presented at the Proceedings of the 2008 International Conference on Advanced Infocomm Technology, Shenzhen, China. https://doi.org/10.1145/1509315.1509324

Zhang, J., Qi, H., Guo, D., Li, K., Li, W., & Jin, Y. (2015). ATFQ: A Fair and Efficient Packet Scheduling Method in Multi-Resource Environments. IEEE Transactions on Network and Service Management, 12(4), 605-617. doi:10.1109/TNSM.2015.2477517

Zhang, T., Gong, T., Han, S., Deng, Q., & Hu, X. S. (2019). Distributed Dynamic Packet Scheduling Framework for Handling Disturbances in Real-Time Wireless Networks. IEEE Transactions on Mobile Computing, 18(11), 2502-2517. doi:10.1109/TMC.2018.2877681

Zhu, J., Zhao, M., Zhang, S., & Zhou, W. (2020). Exploring the road to 6G: ABC — foundation for intelligent mobile networks. China Communications, 17(6), 51-67. doi:10.23919/JCC.2020.06.005

Jialai Wang, Han Qiu, Yi Rong, Hengkai Ye, Qi Li, Zongpeng Li, and Chao Zhang. 2022. BET: black-box efficient testing for convolutional neural networks. In Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2022). Association for Computing Machinery, New York, NY, USA, 164–175. https://doi.org/10.1145/3533767.3534386