AVIRA: Enhanced Multipath for Content-aware Adaptive Virtual Reality

Fabio Silva
Student Member, IEEE
School of Electronic Engineering
Dublin City University, Ireland
fabio.silva3@mail.dcu.ie

Mohammed Amine Togou
Member, IEEE
School of Electronic Engineering
Dublin City University, Ireland
mohammedamine.togou@dcu.ie

Gabriel-Miro Muntean
Senior Member, IEEE
School of Electronic Engineering
Dublin City University, Ireland
gabriel.muntean@dcu.ie

***Abstract*—This paper presents Adaptive VR (AVIRA), a scheme that implements a Virtual Reality (VR) content-aware prioritisation transport to extend Multipath TCP (MPTCP) functionalities and improve its performance. To do so, AVIRA monitors the subflows operation and forecasts subflows' performance by applying an Machine Learning (ML) approach to evaluate a set of features - such as latency and throughput - for every subflow available. This ML approach forecasts the performance of these features through linear regression and applies a linear classifier by using a weighted sum on the forecast results. When the traffic of a specific VR component is detected, AVIRA performs its prioritisation scheme by redirecting packets to the subflow with the best set of forecasted features. AVIRA outperforms the algorithms used for comparison and shows that the use of an ML approach in a "*low-level*" application is viable, especially in situations where the network features under scrutiny are subject to higher variations. In these scenarios, the AVIRA scheme can be outstandingly efficient.**

***Keywords*—machine learning, multipath TCP, regression, virtual reality, network transport improvement, neural network**

I. Introduction

Between 2014 and 2019 [1], [2], HBO® Go streaming services (a subsidiary of WarnerMedia® Entertainment) experienced instabilities and crashed, leaving thousands of US customers off-line during one of the most popular TV shows, *"Game of Thrones"*. These were not isolated incidents but public examples of a pressing matter: the demand for data. As VR expands its popularity beyond the entertainment sector [3] covering a vast range of areas (e.g., education [4], [5] and medical rehabilitation [6]), its stringent demands for resources makes this situation increasingly concerning.

The global IP traffic is expected to increase threefold between 2017 and 2022 and reach 50 GB monthly IP traffic per capita by 2022 [7]. The global number of Machine-to-machine (M2M) connections is also expected to increase to 4.4 billion by 2023 (a four-fold growth) [8]. Although the high level of research (e.g., adaptive multimedia [9] and HD video over wireless networks [10]), the constant investments and the development of new technologies (e.g., 5G [11], Software-Defined Networking (SDN) [12] and Network Function Virtualisation (NFV) [13]), the demand increases even more rapidly.

Alternatives to improve networks performance have been proposed and multipath is among them. Despite all the progress in telecommunication networks, TCP has been the main technology to reliably transfer data for IP networks for nearly 40 years now [14]. However, as applications and networks evolve, some limitations have surfaced (e.g., strict order-of-transmission delivery and socket limited scope limiting data transfer through multi-homed hosts). Several projects addressing multipath technologies with different approaches and intent usage have been suggested (e.g., Stream Control Transmission Protocol (SCTP) [15] and Multipath Quick UDP Internet Connection (MPQUIC) [16]). MPTCP technology in particular (a multipath transport layer protocol that expands TCP to enable data transport over multiple paths [17], [18]) is the object of study in this paper.

Furthermore, ML has been applied to several fields of human knowledge and also used to improve networking performance in a diverse set of approaches [19]. Given the variety of network technologies and levels of complexity, several ML algorithms have been explored to deal with intricate problems and scenarios. Tasks demanding classification, regression or decision-making techniques are candidates for implementing ML algorithms and leveraging networking performance.

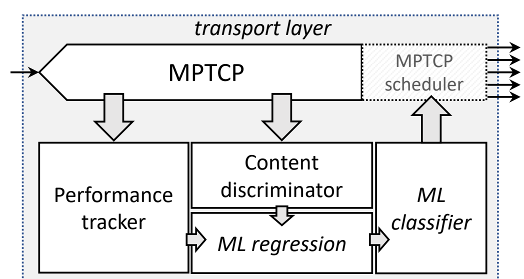


Fig. 1: AVIRA main blocks

This paper combines ML and MPTCP technology to propose AVIRA, an innovative scheme that improves MPTCP management and, concomitantly, monitors VR content traffic to develop a content-aware prioritisation policy. Indeed, AVIRA can detect traffic of specific VR components and redirects it to the subflow with the shortest Round-Trip Time (RTT), based on subflows performance forecast. Figure 1 presents AVIRA's architecture and main components. A more detailed description can be found in Section III.

AVIRA's prioritisation scheme can significantly enhance the performance of many VR applications. For instance, prioritising information such as Inertial Measuring Unit (IMU) can improve the performance of applications that are highly dependent on movement tracking while prioritising video traffic can be key to ameliorate the performance of rich multimedia applications.

The rest of this paper is organised as follows. Section II surveys some related work on ML, MPTCP and VR concepts studied in this paper. Section III describes AVIRA's architecture and the ML algorithm. Section IV details the testbed implementation and assessment results. Finally, Section V concludes the paper and suggests future work.

II. Related works

AVIRA is deployed at the transport layer of the Open Systems Interconnection (OSI) model and sits on the top of the MPTCP protocol. AVIRA monitors subflows' activities and, by applying an ML approach, forecasts subflows' performance based on its features - such as latency or throughput. It also monitors the VR components and redirects specific packets to the subflow that suits them best. This section gives an overview of the main technologies deployed in this research paper.

MPTCP technology *extends* TCP to enable data transport over multiples paths concurrently and transparently [17]. These multiple data flows are named *subflows* and MPTCP ensures they operate with the application layers as regular Transmission Control Protocol (TCP) sessions. This strategy is defined to assure that all subflow connections act as regular TCP session/connections, preserving compatibility by keeping the interaction between OSI layers unaffected. This avoids interference of *middle-boxes* [17], [18] (such as firewalls and routers) and facilitates the adoption of MPTCP technology.

Many studies show that MPTCP can be optimised for specific applications. For example, in [20], the authors studied the impact of path selection in MPTCP performance. Their results indicate that MPTCP congestion control algorithms do not *obviate* the necessity of a better packet scheduler. Their considerations about congestion control, buffer size and an RTT-aware scheduler indicate that it is still necessary a scheduler capable of improving MPTCP's performance - even if for a specific use, which is the goal of this work: applying ML techniques to predict MPTCP subflows' performance.

The use of ML in networking is not something new. Indeed, ML techniques have been widely applied to several types of networking applications, e.g., job scheduling and traffic prediction [19]. They are also deployed at various levels of the networking model, from a higher-level definition of the geographic distribution of data centres to lower level forecast of network traffic. The focus of this paper is the lower levels, namely, the TCP transport layer.

In [21], the authors propose a transport layer targeting distributed machine learning. Nonetheless, the proposed approach can lead to a long tail flow completion time and is not practical for real-time applications. In [22], a congestion control scheme "samples" the network for a period of time and collects results for ulterior analysis. In [23], the authors described a throughput prediction scheme to select an optimised initial bitrate for streaming adaptation. A similar approach was presented in [24] which analyses historical samples of networking features and uses them to help define a more adequate TCP congestion size. Despite having a similar logic, there are distinct differences, in terms of methodology, when comparing AVIRA to the aforementioned approaches. Although [22], [23], [24] sample or measure "*low level*" network features, they use off-line analysis and training mechanisms to implement their proposed solutions. As described in details in Section III, rather than using massive training data or large amounts of sample data, AVIRA uses a reduced data set sampled in real-time and applies simple ML algorithms to process such information.

III. AVIRA architecture and algorithm

AVIRA evaluates MPTCP subflows' performance and alters their management to promote a content-aware prioritisation scheme targeting the *motion-to-photon* problem: the time between the detection of a movement or action until this interaction reflects the user's experience. As detailed in [25] work, in order to create an immersive VR experience, several distinct components are involved. Figure 2 depicts how some of these components interact to create the virtual experience and how the interdependency between them creates the motion-to-photon problem.

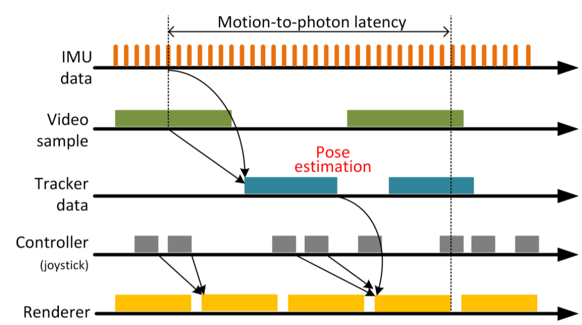


Fig. 2: The motion-to-photon problem.

This is the reason why it is important to deliver a content-aware functionality to a VR application (or any other application demanding a prioritisation scheme) to improve how these components are *assembled* together. To achieve this, AVIRA implements the following:

A. Performance tracker

Monitors constantly subflows' feature performance (e.g., throughput and/or latency) and creates a performance history buffers for the operation. These history buffers are then used to extract a forecast about features' behaviour through the application of linear regression algorithms.

B. Content discriminator

To promote a content-aware prioritisation policy, packets are monitored constantly and, when specific traffic is identified (i.e., by checking the packet headers), it is redirected to the subflow that suits it the most (i.e., subflow providing the highest throughput or shortest RTT).

C. Machine learning algorithms

AVIRA's ML algorithms description adopts a simplified workflow described in [19] and summarises briefly how they are applied here:

- *Problem*: subflows' performance must be forecasted in real-time using a reduced time-series to avoid memory and/or processing overhead.
- *Data*: MPTCP subflow pool monitors and captures the subflows' behaviour in real-time to create a performance history data in a time-series format.
- *Analysis*: the algorithms will extract specific features for analysis and performance prediction.
- *Model*: first, a linear regression slope is used to predict a feature's behaviour in every subflow of the pool and a linear classifier is used afterwards to choose the most suitable subflow.
- *Validation*: AVIRA is compared to other algorithms (see Section IV) with different approaches.
- *Deployment*: AVIRA acts as an MPTCP scheduler in the [26] implementation used for tests and assessment.

D. Scheduler

The MPTCP scheduler behaviour is altered, based on the results of the ML analysis, to deliver the best performance in the next operation.

It is important to emphasise that AVIRA operates while observing a few limitations concerning the ML computation. As AVIRA captures and processes data in real-time, it is mandatory to have a small footprint or processing overhead. Unlike other algorithms which use off-line analysis and training data [19], AVIRA does not support heavy data sets for both training or performance evaluation. The dataset used is a time-series composed by a key-value pair that represents the feature value for a given packet transport (e.g., at *00:00:11s* the packet had a throughput of 1Mb).

Also, AVIRA employs a simplified linear regression for performance forecast and a quick linear classifier to choose between the subflows. As it can be seen on Algorithm 1, the implementation of those algorithms are straightforward and poses no heavy computational overhead, which is important for a process supposed to run in real-time. These ML algorithms are detailed in the following subsections:

E. Linear regression

As a way to forecast subflows' performance, AVIRA uses a simple linear regression - which is a type of regression analysis commonly used in machine learning algorithms based on supervised learning. It means that for each feature analysed (RTT, throughput, etc.) both the input variable x (or independent variable) and output variable y (or dependent variable) are known and there is a linear relationship between x and y, as shown in Equation 1a. Simply put, the purpose of the linear regression algorithm is to compute the values for b_0 and b_1 that best represent the linear relationship between the independent and dependent variables. Equation 1b and Equation 1c show how the values for b_0 and b_1 are obtained.

$$y = b_0 + b_1 x \tag{1a}$$

$$\begin{aligned} b_0 &= \left(\sum_{i=1}^{n} y_i\right)\left(\sum_{i=1}^{n} x_i^2\right)-\left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} x_i y_i\right)/n\left(\sum_{i=1}^{n} x_i^2\right)-\left(\sum_{i=1}^{n} x_i\right)^2 \\ &= \overline{y}\left(\sum_{i=1}^{n} x_i^2\right)-\overline{x}\left(\sum_{i=1}^{n} x_i y_i\right)/\left(\sum_{i=1}^{n} x_i^2\right)-n\overline{x}^2 \end{aligned} \tag{1b}$$

$$\begin{aligned} b_1 &= n\left(\sum_{i=1}^{n} x_i y_i\right)-\left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)/n\left(\sum_{i=1}^{n} x_i^2\right)-\left(\sum_{i=1}^{n} x_i\right)^2 \\ &= \left(\sum_{i=1}^{n} x_i y_i\right)-n\overline{x}\overline{y}/\left(\sum_{i=1}^{n} x_i^2\right)-n\overline{x}^2 \end{aligned} \tag{1c}$$

Where x is the independent variable (input), y is the dependent variable (or output), b_0 is the y-intercept (Equation 1b) and b_1 is the slope of the function (Equation 1c).

In practical terms, the x variable is a timestamp representing the exact moment a packet is transported and the y variable is the value of a specific network feature. These variables are the components of the dataset used in all linear regression forecasts. By forecasting how a specific subflow feature will behave in the next interaction, a linear regression can help the MPTCP scheduler choose the best subflow available in the subflow pool. To illustrate how it chooses between subflows, Figure 3 shows a hypothetical situation where a feature (e.g., latency) is monitored in a group of subflows.

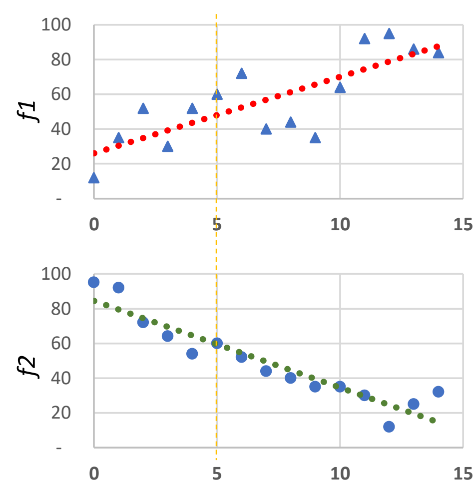


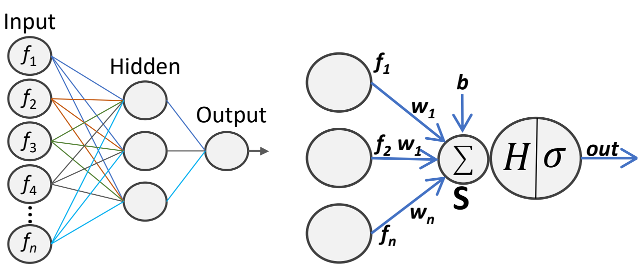
Fig. 3: Regression-based decision example

During operation, AVIRA has to decide which subflow has the best chances of offering the best performance (in this case, the lower latency). In the example in Figure 3, the moment of decision is represented by the dotted yellow line at $t = 5$. Although the last measured values are *60ms*, the linear regression (dotted line in each plot) forecasts a smaller latency for f_2 when compared to f_1 for the next interaction and, consequently, f_2 has higher chances to offer lower latency for the next operation.

F. *Linear classifier*

Once the subflows' features are analysed, AVIRA applies a classification algorithm to "*combine*" the subflow features using a weighted constraint array to evaluate the subflows considering multiple features. To do so, AVIRA examines the subflows using a linear classifier approach, which models the trigger boundary based on a linear combination of inputs.

Figure 4 depicts the subflow pool and its analysis using an Artificial Neural Network (ANN) notation. By this notation, the linear classifier in Figure 4(a) considers the subflow pool features as the provider of the input layer and the analysis combining the subflow features and weighted constraints as a hidden layer.



(a) Features as an ANN (b) Subflow as a neuron

Fig. 4: Linear classifier representation

In Figure 4(b), the subflow *features* and their *weights* are represented by vectors, $\vec{f}$ and $\vec{w}$ respectively. The *dot product* (or weighted sum) is applied to both vectors (Equation 2). Usually, $\dot{v}$ performs as a threshold (or activation) function computing the weighted sum ($\vec{w} \times \vec{f}$) into a single result. This activation function $H(\dot{v})$ (Equation 3) defines whether the neuron would be fired or not: the value is 1 if the analysed conditions satisfy the weighted sum validating these conditions or 0 if these conditions are not enough to *classify* a specific neuron eligible as a solution. This is achieved by applying an activation function as shown in Equation 4.

$$y = \dot{v}(\vec{f} \times \vec{w}) = (\sum_{i=1}^{n} f_i \cdot w_i) \tag{2}$$

$$H(\dot{v}) = H(\vec{f}.\vec{w}) = \begin{cases} 1, & \text{if } v <= 0 \\ 0, & \text{if } v > 0 \end{cases} \tag{3}$$

$\vec{f}$ is the subflow's features forecast in the previous process and $\vec{w}$ are parameters used to adjust the *relevance* of each feature and offers a mean to adapt to a specific domain or applicability. $H(\dot{v})$ is a Heaviside step function achieved by a *Sigmoid* activation function defined in Equation 4.

$$\sigma(\dot{v}) = \frac{1}{1 + e^{-\dot{v}}} = \frac{1}{1 + e^{-(\vec{f}.\vec{w})}} \in (0, 1) \tag{4}$$

G. AVIRA *Algorithm*

Algorithm 1 describes the pseudocode for both linear regression and linear classifier when a prioritised packet is detected. First, the algorithm recovers the array used to store the subflows' performance buffer (P_B) and calculates the linear regression (or forecast) of every feature stored there.

Algorithm 1: Linear regression and classifier usage.

```
Result: Forecast subflow
Input: P_B ← subflow's history performance buffer.
       W_L, W_T; //weighted constraints (latency/throughput).
V_p; // array of a feature from all subflows.
V_L; // array of a linear classifier results.
//computes linear regression for all subflows
foreach (V_p in P_B) do
    foreach feat in V_p do
        n++;
        S_x += feat.time;
        S_y += feat.value;
        S_xy += (feat.time * feat.value);
        S_x2 += pow(feat.time, 2);
    end
    ȳ = S_y / n;
    x̄ = S_x / n;
    b_0 = ((ȳ * S_x2) - (x̄ * S_xy)) / (S_x2 - (n * pow(x̄, 2));
    b_1 = (S_xy - (n * ȳ * x̄)) / (S_x2 - (n * pow(x̄, 2));
    y = b_0 + b_1 * current_time;
    V_p.perf = y; //update performance of a subflow
end
//computes linear classifier
L_v ← P_B.getLatency();
T_v ← P_B.getThru();
for i ← 1 to P_B.pool.size do
    V_L[i] ← (L_v[i] * W_L[i]) + (T_v[i] * W_T[i]);
    V_L[i] ← (1 / (1 + (1/pow(e, V_L[i])));
end
sort(V_L);
return V_L.get(0);
```

After that, the algorithm recovers specific features (in this case, latency L_v and throughput T_v) and combines them with the respective weighted constraints (W_L and W_T) using a linear classifier to identify the subflow that offers the best performance for the next operation.

IV. Testbed and Assessment

AVIRA uses a Network Simulator 3 (NS-3) open-source MPTCP implementation [26] of the IETF RFC 8684 [17] to implement, test and assess its simulation environment. The simulation scenario consists of a Point-to-Point (P2P) model with nodes having 1Mbps bandwidth rate and 2ms default latency. A MpTcpBulkSender and a MpTcpPacketSink application, configured in the source node (n_0) and the sink node (n_4) respectively, extend the original NS-3 implementations and are set up to send and receive data in a "bulk copy" style, i.e., sending a large amount of data as fast as possible over the MPTCP. Both source and sink nodes are configured in single-homed mode, i.e., each node has one

NS-3 device. These nodes (source and sink) are connected through other three intermediary devices to create two different pathways and 8 subflows are set using different ports. For this implementation, NS-3 defines automatically the routing tables for establishing the traffic flow among these nodes. A summary of the simulation environment is presented in Table I.

The simulation scenario implements a prioritisation scheme where AVIRA identifies specific packets (content discriminator) to trigger the ML processes (linear regression and linear classifier) to evaluate and identify which subflow would transmit the packet with the best performance. AVIRA assumes that one packet out of 500 packets (1/500) is the average between a *simpler* VR component (e.g., gyroscope or accelerometer data) and more complex components (such as video data packets) [27].

For assessment purposes, AVIRA is compared to three algorithms: 1) the default MPTCP used in NS-3, labelled as *Default*; 2) a Random Scheduler, labelled as *Random*; and 3) a *lowest* RTT approach suggested by [28], labelled as *Lowest*. Figure 5 illustrates how the traffic is distributed between the different subflows (0 to 7) available in the MPTCP subflow pool during the simulation.

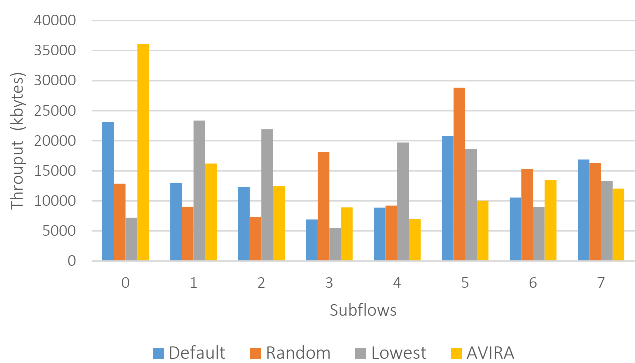


Fig. 5: Traffic distribution: algorithm vs subflow

Despite the algorithms having a slightly different final traffic volume, due to eventual differences in the control information, the same cannot be said about the distribution between the subflows. When comparing AVIRA with the remaining algorithms, it is evident that AVIRA allocates the traffic in a more *concentrated* way. This concentration is mostly defined by the forecast scheme during the prioritisation of packets - eventually, the congestion control refuses the redirection to a specific subflow due to the size of the available TCP window.

TABLE I: Simulation setup

Parameter	Value
Environment	NS-3 open source MPTCP [26]
Simulation length	1200 seconds
Number of nodes	5 Nodes
Data Rate	1Mbps
Delay	2ms
Number of subflows	8
Prioritisation ratio	1/500
Sender app	MpTcpBulkSender [26]
Receiver app	MpTcpPacketSink [26]

Figure 6 describes how a network feature (in this case, the RTT) was impacted during the simulations (for RTT, the smaller the values the better). AVIRA had, on average, smaller RTT values when compared to the other three algorithms. Table II shows the average value for each algorithm and its standard deviation. However, AVIRA has a more impacting performance when, in some cases, AVIRA delivers RTT that is 30% lower than the default MPTCP algorithm. This peak performance is present especially in situations when the overall RTT shows higher variation due to some network conditions. Also, AVIRA generates a throughput gain of 3.4%.

TABLE II: AVIRA assessment average values

Default		Random		Lowest		AVIRA	
RTT	SD	RTT	SD	RTT	SD	RTT	SD
927	83.6	908	91.0	915	92.0	893	89.2

RTT in milliseconds; SD: Standard Deviation.

During the assessment, other measurements were examined to guarantee that no unintentional side effects were caused by the various implementations of the testbed. To this end, some TCP protocol features were monitored to confirm that the regular operation was not impacted. The traffic presented in Figure 5 showed no significant deviations. Additionally, features such as number of retransmissions, duplicate ACK, lost segments or fast retransmissions offered no significant degradation [29], varying between 0.07% and 0.09% for retransmissions and duplicate ACK, respectively.

V. Conclusions

By evaluating the possible impact of various types of schedulers on MPTCP performance, our results indicate that using AVIRA as an optimised scheduler, tailored specifically for MPTCP prioritised delivery of VR components, can bring significant contribution for improving VR Quality of Service (QoS). Simulation results show that AVIRA outperformed the existing algorithms. This indicates that the use of an ML approach in a "*low-level*" and real-time implementation (transport layer) is viable especially in situations where the network features under scrutiny are subject to higher variations. In these scenarios, AVIRA can be outstandingly efficient.

Future investigations should address an adaptive algorithm combining ML algorithms and content prioritisation scheme for VR components and develop a study correlating QoS improvements (or deterioration) and its impact on Quality of Experience (QoE).

Acknowledgements

This work was supported, in part, by the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 688503 and, in part, by Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero - the Irish Software Research Centre (www.lero.ie).

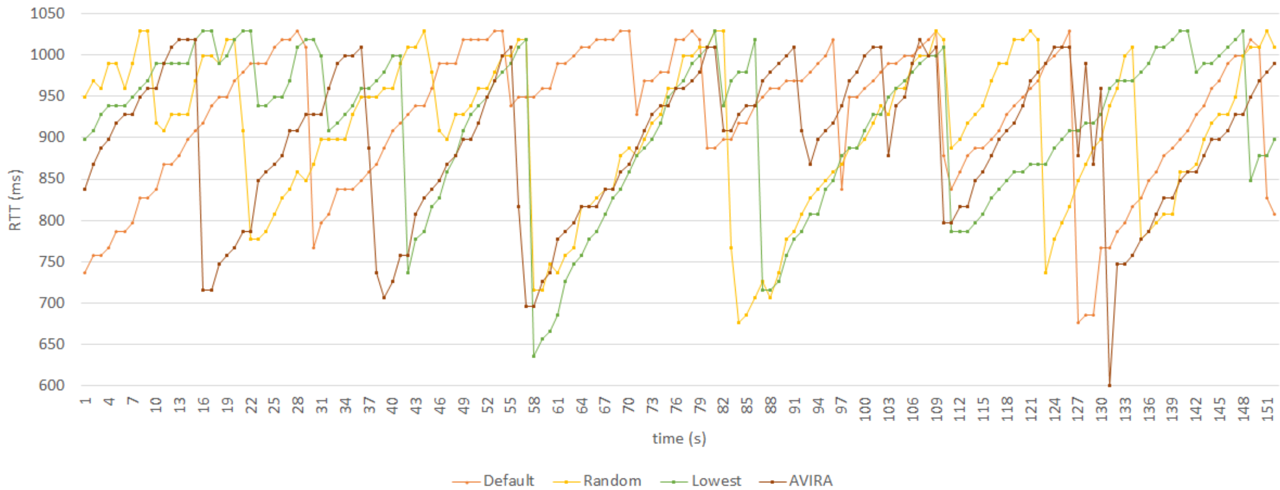


Fig. 6: AVIRA Assessed against MPTCP default, Random and Lowest algorithms.

REFERENCES

[1] C. Campbell, "HBO Go Crashes During Game of Thrones Season 4 Premiere," 2014. [Online]. Available: http://bit.ly/37VrZc3

[2] K. Wynne, "Is HBO Go, HBO Now Down Or Not Working?" 2019. [Online]. Available: http://bit.ly/394StrK

[3] R. T. Azuma, "A survey of augmented reality," *Presence by MIT*, vol. 6, no. 4, pp. 355–385, 1997. [Online]. Available: http://bit.ly/2NBMcKJ

[4] D. Bogusevschi, C. Muntean, and G.-M. Muntean, "Teaching and Learning Physics using 3D Virtual Learning Environment: A Case Study of Combined Virtual Reality and Virtual Laboratory in Secondary School," *Journal of Computers in Mathematics and Science Teaching*, vol. 39, no. 1, pp. 5–18, 2020.

[5] C. H. Muntean, D. Bogusevschi, and G.-M. Muntean, *Innovative Technology-based Solutions for Primary, Secondary and Tertiary STEM Education*. Paragon Publishing, 2019.

[6] C. Kirner, C. Cerqueira, and T. Kirner, "Using Augmented Reality Artifacts in Education and Cognitive Rehabilitation," *Virtual Reality in Psychological, Medical and Pedagogical Applications 2 Will-be-set-by-IN-TECH*, pp. 247–270, 2012.

[7] Cisco VNI Mobile, "Cisco visual networking index (VNI) global mobile data traffic forecast update, 2017-2022 white paper," Cisco Inc., Tech. Rep., 2019. [Online]. Available: http://bit.ly/2XdLc3H

[8] Cisco, "Cisco Annual Internet Report (2018–2023)," *Cisco*, pp. 1–41, 2020. [Online]. Available: https://bit.ly/3bVPJyc

[9] M. Wang, C. Xu, X. Chen, L. Zhong, and G. M. Muntean, "Decentralized asynchronous optimization for dynamic adaptive multimedia streaming over information centric networking," *Journal of Network and Computer Applications*, vol. 157, no. 102574, 2020.

[10] T. Casey and G. M. Muntean, "Delivery of High Definition Video Content over Bandwidth Constrained Links in Heterogeneous Wireless Networks," in *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, BMSB*, 2019, p. 7.

[11] NGMN Alliance, "NGMN 5G White Paper," *Next Generation Mobile Networks, White paper*, pp. 1–125, 2015. [Online]. Available: http://bit.ly/2tkosF7

[12] M. A. Togou, D. A. Chekired, L. Khoukhi, and G.-M. Muntean, "A Hierarchical Distributed Control Plane for Path Computation Scalability in Large Scale Software-Defined Networks," *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 1019–1031, 2019.

[13] D. A. Chekired, M. A. Togou, L. Khoukhi, and A. Ksentini, "5G-Slicing-Enabled Scalable SDN Core Network: Toward an Ultra-Low Latency of Autonomous Driving Service," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 8, pp. 1769–1782, 2019.

[14] DARPA, "Transmission Control Protocol - RFC 793," Information Sciences Institute - University of Southern California, Tech. Rep. September, 1981.

[15] Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, "RFC 4960 - Stream Control Transmission Protocol," 2007. [Online]. Available: https://tools.ietf.org/html/rfc4960

[16] Q. D. Coninck and O. Bonaventure, "Multipath Extension for QUIC," 2017. [Online]. Available: http://bit.ly/2ueEZuE

[17] A. Ford, C. Raiciu, M. Handley, U. Bonaventure, and C. Paasch, "RFC 8684: TCP Extensions for Multipath Operation with Multiple Addresses," Internet Engineering Task Force (IETF), Tech. Rep., 2020.

[18] H. Liu, Q. Li, B. Wu, Q. Wu, J. Zhang, and J. Zhou, "Trusted Multipath-TCP (MPTCP) extension," Internet Engineering Task Force (IETF), Tech. Rep., 2019.

[19] M. Wang, Y. Cui, X. Wang, S. Xiao, and J. Jiang, "Machine Learning for Networking: Workflow, Advances and Opportunities," *IEEE Network*, vol. 32, no. 2, pp. 92–99, 2018.

[20] B. Arzani, A. Gurney, S. Cheng, R. Guerin, and B. T. Loo, "Impact of path characteristics and scheduling policies on MPTCP performance," *Proceedings - 2014 IEEE 28th International Conference on Advanced Information Networking and Applications Workshops, IEEE WAINA 2014*, pp. 743–748, 2014.

[21] J. Xia, G. Zeng, J. Zhang, W. Wang, W. Bai, J. Jiang, and K. Chen, "Rethinking transport layer design for distributed machine learning," *ACM International Conference Proceeding Series*, pp. 22–28, 2019.

[22] M. Dong, Q. Li, D. Zarchy, P. B. Godfrey, and M. Schapira, "PCC: Re-architecting congestion control for consistent high performance," *Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2015*, pp. 395–408, 2015.

[23] Y. Sun, X. Yin, J. Jiang, V. Sekar, F. Lin, N. Wang, T. Liu, and B. Sinopoli, "CS2P: Improving video bitrate selection and adaptation with data-driven throughput prediction," *SIGCOMM 2016 - Proceedings of the 2016 ACM Conference on Special Interest Group on Data Communication*, pp. 272–285, 2016.

[24] K. Winstein and H. Balakrishnan, "TCP ex Machina: Computer-Generated Congestion Control," *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM - SIGCOMM '13*, 2013.

[25] D. Wagner, "Motion-to-photon Latency in Mobile AR and VR," 2018. [Online]. Available: https://daqri.com/blog/motion-to-photon-latency/

[26] M. Kheirkhah, I. Wakeman, and G. Parisis, "Multipath-TCP in ns-3," in *Cornell University Library*, 2015, pp. 3–4. [Online]. Available: https://bit.ly/2MfiElc

[27] M. Chouiten, J.-Y. Didier, and M. Mallem, "Distributed Augmented Reality Systems: How Much Performance is Enough?" *2012 IEEE International Conference on Multimedia and Expo Workshops*, pp. 337–342, 2012.

[28] C. Paasch, S. Barré, F. Duchêne, and G. Detal, "Multipath TCP in the Linux Kernel," 2019. [Online]. Available: https://www.multipath-tcp.org

[29] J. S. Mwela and O. E. Adebomi, "Impact of Packet Loss on the Quality of Video Stream Transmission," Ph.D. dissertation, Blekinge Institute of Technology, 2010.