IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 1 August 2022, accepted 4 September 2022, date of publication 8 September 2022, date of current version 16 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3205418

RESEARCH ARTICLE

NexGen S-MPTCP: Next Generation Smart Multipath TCP Controller

SUKHDEEP SINGH[1], (Member, IEEE), MUKESH KUMAR MAHESHWARI[2], MADHAN RAJ KANAGARATHINAM[1], (Senior Member, IEEE), AURANGZEB RASHID MASUD[2], AND IQBAL QASIM[3]

[1]Samsung R&D Institute India Bangalore, Bengaluru 560037, India
[2]Department of Electrical Engineering, Bahria University Karachi Campus, Karachi 75260, Pakistan
[3]Department of Computer Science, University of Science and Technology, Bannu 28100, Pakistan

Corresponding author: Mukesh Kumar Maheshwari (mukeshkumar.bukc@bahria.edu.pk)

ABSTRACT A consistent increase in internet streaming via extensive surfing, online gaming and audio-visual streaming has posed many challenges in maintaining the quality of service for the communication industry. To cater the issues of Next Generation Mobile Networks previously, we proposed a Smart MPTCP (S-MPTCP) path controller that handles the MPTCP sub-flows while having various network communication interfaces. The proposed scheme controls the MPTCP sub-flows utilizing the information exchanged at the start of establishing a connection to appropriately map the client with a suitable network server interface. Here, we extended our previous work by analyzing it in 5G networks and over video traffic. To determine its efficacy, the proposed scheme was simulated at Samsung Electronics, Head Quarters and validated through live air experiments using Samsung Galaxy S21 and Galaxy Note 20 devices with leading Korean Telecom operators (KT and SKT) connection. Our simulation outcomes and live air experiments have proved that the proposed scheme can efficiently control the MPTCP sub-flows with enhanced throughput and nominal connection overhead. The previously proposed S-MPTCP was tested on Samsung Galaxy S8 and Galaxy Note 8. We have re-tested the LTE network case on latest Samsung devices i.e. Samsung Galaxy S21 and Galaxy Note 20. In the case of LTE network, the battery power gain improved by 13% and approximately 50% of mobile data saved compared to the existing conventional MPTCP approach. The gains observed using new devices are same as in old devices. We have taken a step further to test our solution in 5G NR conditions. In the case of 5G NR for good WiFi, the average utilization of 5G Data with S-MPTCP is 27%, whereas it is 53% without S-MPTCP.

INDEX TERMS MPTCP, path manager, data saving, power saving, smartphones.

I. INTRODUCTION

The huge influx of smart devices in society foresee a connection of around 18 billion IPV6 devices with the network by the year 2022. Almost half of the 18 billion IPV6 devices are expected to be the IPV6 cellular mobile devices alone [1]. Recently, the use of smart phones has gained an immense momentum along with a tremendous increase in network traffic. It has been observed that streaming services especially multimedia oriented data streaming has become the major proportion of this traffic and its popularity is surging on daily basis. Owing to this fact, it is expected that by the end of year 2022, 82% of the world's IP and smartphone traffic will be solely in the form of video streaming and that will account a rate of 26, 100 MBs per month [1]. The transport layer protocol design faces challenges because of the huge mobile network traffic and fluctuating network bandwidth that results in the decline of network condition. The general typical problems that media streaming services may encounter as a result of this could be extended startup delays, poor switching quality and frequent stalls. With the massive development of intelligent hardware, by 2022 network

The associate editor coordinating the review of this manuscript and approving it for publication was Zesong Fei.

society aims to connect 18 billion IPV6 devices out of which 9.4 billion IPV6 mobile devices [1]. We have tremendous momentum around Smartphones.

Among all transport layer protocols, TCP is the most widely used protocol for robust data transmission with higher efficiency [2]. However, recently an enhanced version of TCP called as Multipath TCP (MPTCP) is introduced. MPTCP has become a propitious standard to combat the challenges faced by the network providers in the provision of quality streaming services for satisfactory user experiences. MPTCP is standardized in the Internet Engineering Task Force (IETF) RFC 6182 [3]. It enables the operations of TCP over multiple paths with multiple addresses. MPTCP utilizes the multipath operation at transport layer level, it ensures the reliability and increases the capacity of the network for a smooth and seamless operation [3], [4]. MPTCP is considered as a decent solution for smart devices facing inconsistent signal quality and coverage issues from network interfaces like LTE-A and WiFi. Although, various network operators have already deployed the MPTCP to meet the user requirements by improving the service quality and throughput of the network, still there remains a considerable space for improvement in the mobile data management and power saving domains.

MPTCP is resilient to link failures and can effectively use the available bandwidths. It uses TCP in its sub flows and improves the throughput by aggregating the capacity. The MPTCP sub flows are established over multiple links. However, in case of unreliable connection the sub flow connections tend to break [8]. The use of multiple network interfaces adheres a trade-off between the throughput and power consumption of the device. Contrary to the wired users, the user of smart phones avoids to utilize all the accessible interfaces, particularly the use of cellular interfaces such as LTE, as it may burden the user with additional data charges. A number of studies are carried out by the researchers to improve and optimize the performance of MPTCP for different scenarios [2]. However, these studies do not satisfy the user requirement for managing simultaneous multiple network communication interfaces utilization. These limitations demand a smart multipath controller that efficiently takes the decisions and controls the sub flows to enhance the reliability, data and power savings keeping the throughput intact.

To fulfill the Next generation Mobile Networks demands and to combat the challenges faced by MPTCP, we introduce a novel Smart MPTCP controller (S-MPTCP) in our previous work [5]. The S-MPTCP enables the analysis, classification and the management of the MPTCP sub-flows. Our proposed S-MPTCP concept enables the efficient controlling of MPTCP sub-flows on the basis of path characteristics and interfaces. In [5], we proposed a MPTCP path controller for LTE-A networks. However, previous work lacks in analysis and performance evaluations of SMPTCP over 5G NR and video. The major contributions of our research presented in this paper are enumerated as:

1) In this work, we propose and analyze the S-MPTCP architecture for varying channel condition in 5G mmWave network by considering the video traffic. The SMPTCP ensures the best utilization of network condition and route the traffic to the best available path to reduce the data and power consumption.
2) To analyze the viability and effectiveness of our proposed concept, we performed rigorous simulations in our lab at Samsung Electronics, Head Quarter South Korea, and verified the simulation results with experimental results. The live air experiments were carried out using Samsung Galaxy S21 and Galaxy Note 20 devices with the connections of KT and SKT South Korean operators.
3) The simulation and live air experiment results confirm that the proposed approach can efficiently control and manage the MPTCP sub-flows. Simultaneously, it delivers a better throughput with nominal connection overheads. This improves the battery power gain by 13% and hence saves the mobile data consumption up to 50% in contrast to the currently existing MPTCP solutions.

The initial version of this work is published in [5], where we analyze the MPTCP path controller for LTE-A networks. However, this paper extends the previous published work [5] in several aspects, as mentioned below:

1) Section II provides the summary of the recent works in MPTCP along with the objective of each work referred.
2) In Section III we have provided a method for fine grain traffic classification for MPTCP. Using the classification, the video traffic is detected and we provided a novel algorithm to handle the video traffic in MPTCP.
3) In section IV, the LTE experiments have been re-performed on the latest Samsung devices to re-confirm the gains that were witnessed in published work.
4) In section IV, we had added new results for 5G New Radio (NR) considering varying channel condition. Moreover, Table. 1, Fig. 5 and Fig. 9 are new and unpublished.

The organization of the rest of the manuscript is as follows: Section II provides a short review of the related work on the present legacy MPTCP solutions. Section III explains the proposed scheme of S-MPTCP system and algorithms. Section IV is dedicated to demonstrate the evaluation of live air and simulation results while the outcomes and conclusions of the research are delineated in Section V.

II. RELATED WORK

For years researchers are striving to enhance the MPTCP performance related to the network path and congestion control. Some of the research investigations are performed to identify the limitations of MPTCP because of the absence of route manager between the source and destination. The routing mechanism is network type dependent. Beyond one network interface everything related to the data flow depends

TABLE 1. Related Works in Multipath TCP (MPTCP).

Ref.	Work Summary	Objective(s)
[2]	FSA-MPTCP: A cross-layer solution for MPTCP Optimal strategy for different data flow	How MPTCP works with big data
[8]	QoE-driven MPTCP-based data delivery in MANETs Hidden Markov model-based optimal-start multipath routing	Mutipath Routing and MPTCP based data delivery over MANETs
[11]	AMTCP: Dynamically adjusts the subflows according to application workloads	Reduce resource & scheduling overhead, and achieve a higher throughput
[16]	BBR Coupled congestion control algorithm for MPTCP, Propose an adaptively redundant and packet-by-packet (AR&P) scheduler	Boost MPTCP performance with BBR
[17]	Applicable to path with non-homogenous properties	Discrete-time model of the data exchange process in the MPTCP
[18]	Uses MPTCP+IPv6 to support multi-homing for resilience and multi-path transport for load balancing	Live video streaming platform for future networks
[19]	Optimizing the QOE subject to available resources Proposed an online algorithm to address the bandwidth prediction errors	Preference-aware multipath video streaming algorithm over HTTP using MPTCP
[20]	Proposed a path scheduler to reduce the number of path switches	Analyze the impact of path switching on multipath video streaming
[21]	Forward the data unacked packets to non-congested path from the contested path	Analyze the proposed scheme through experiments on our real data center testbed
[5]	Proposed Smart MPTCP to manage and control the sub flows, to mitigate the challenges of MPTCP	Analyze the proposed scheme for LTE-A network

on the implemented routing mechanism of the other network. Another research area identified by several researchers is Adaptive management of MPTCP. Recent studies proposes mobility-aware seamless handover method based on MPTCP [6] and WAN-aware MPTCP in terms of resilience, that minimize the application performance under WAN link failures [7]. A brief summary of the publications related to the Adaptive MPTCP (AMPTCP) is given in table 1.

Ivan Gojmerac, *et al.* presented the concept of Adaptive Multipath routing algorithm for the dynamic management of autonomous systems [9].The suggested algorithm is capable of balancing the traffic load locally at the expense of minimum overhead. Although, the improvements in conventional MPTCP is depicted through simulation results, traffic load balancing among neighboring networks is carried out without considering the whole network that results in the degeneration of overall routing performance.

Coudron, *et al.* [11] also favored the concept of AMPTCP, that dynamically sets the quantity of sub-flows on the basis of application work-load. It first divides the total duration into small intervals to quantify the throughput of each sub-flow for the most recent interval, then the number of sub-flows is set dynamically. This yields in higher throughput for large sub- flows with minimum resource and scheduling overheads for small sub-flows. AMPTCP achieves a reduction in the number of average sub-flows by 37.5% keeping the same throughput as legacy MPTCP. However, the idea lacks in the efficient path routing and controlling mechanism to eradicate the possibility of network congestion, as, it may cause a performance decline even in the small sub-flows.

Another concept suggests the techniques to improve the formation of MPTCP sub-flows so that it may create enough amount of sub-flows considering the diversity of the fundamental path identified by the Location/Identifier Separation Protocol (LISP) [10]. Moreover, a sub-flow cross-layer coordination module is built between LISP - MPTCP by the researchers. Although, the idea declares a remarkable improvement in cloud communication between multi-homed data centers and users, it fails in determining the most suitable routes and numbers of sub-flow if compared with AMPTCP.

Sandri *et.al.* [12] exploited the utilization of MPTCP in Open Flow network to demonstrate the multi flow design. The objective of this design is to improve the throughput of networks facing congestion due to sharing. This topology sends the sub-flows of single MPTCP connections via multiple network routes. The very first sub-flow is fixed to transmit over the path having closest destination, while rest of the sub-flows are directed over randomly selected paths. All sub-flow paths are decided at the beginning of the process which never updates subsequently despite having changes in the network. This consequently leads to an inefficient routing mechanism because of the packet drops.

The dynamic inclusion and exclusion of MPTCP paths following SDN approach is another way to rectify the poor performance issues caused by the enormous defective packets in situations when the bandwidths and delays of paths are diversified [13]. The idea is to dynamically identify the connected path capacity on the basis of continuously varying network conditions and then selecting the suitable paths. Mininet platform was used to perform the tests to investigate the effectiveness of proposed idea. Although, the results showed some improvement prospects in the Quality of Experience (QoE) and higher download speed for adaptive rate video streaming, just adjusting the number of paths in network connections without taking into account the choice of finest path may result in the deterioration of efficient throughput utilization with increased cost overheads.

Zannettou *et al.* depicted the SDN controller assisted MPTCP sub-flows routing scheme [14]. The proposed scheme relies on the packets inspection to determine the suitable MPTCP sub-flow paths and reserves the selected path using two tables. It was found that when sender access links do not impose any restriction or limitations on the sub-flows, MPTCP performance exhibits significant improvement. However, the lack of adaption capability and having an inherent static nature of path selection mechanism limits the MPTCP optimal performance attainment.

X. Yitao, *et al.* [2] presented the cross-layer design for Flow-Size-Aware MPTCP (FSA-MPTCP) concept that determines the upper layer data flows information size to guarantee an adaptable optimal transmission of various data flows. This mode of data transmission is optimized for Quality of Service (QoS) and enhanced data flows. On the other hand, Zhang *et al.* [8] introduced the QoE optimized multipath TCP oriented data delivery system using mobile adhoc networks. The realization of multipath data routing with optimized start was realized using hidden Markov model. The performance evaluations show that the proposed algorithm utilizes multiple sub-paths more efficiently with finer network traffic load balancing attribute. Han *et al.* in [16] proposed the novel Bottleneck Bandwidth and Round-trip time (BBR) congestion control algorithm for MPTCP and Adaptively Redundant & Packet-by-Packet (AR & P) scheduler. The presented algorithm was able to maintain stability in lossy network situations with higher throughput, balanced congestion and assured fairness. In [17], Igaciuk *et al.* proposed a discrete-time data exchange model accompanied with a non-linear sliding mode MPTCP controller.

Later, the authors in [18], [19], and [20] introduced MPTCP performance enhancement for video streaming purpose. In [18] Luo *et al.* proposed a real time video streaming platform for future mobile edge computing scenarios using IPv6+MPTCP. In [19] preference aware adaptive algorithm is presented for video streaming over S-MPTCP. The algorithm efficiently predicts the bandwidth error and optimizes the QoE subjected to the available resources. In [20] the impact of path switching on multipath video streaming was studied. The authors proposed a path scheduler and utilized the network performance measures and video quality metrics for overall performance analysis.

All the proposed MPTCP based routing mechanisms available in literature lacks in the simultaneous handling of sub-flows congestion, dynamic network conditions and network packet drops. This gap provided us the basis of designing an intelligent MPTCP sub-flows path controller that enables an efficient handling of all the above mentioned MPTCP related issues along with an improved user experience in terms of enhanced throughput and minimum data and power consumption. The proposed S-MPTCP manages and controls the sub-flows on the basis of path characteristics and interfaces. The performance of S-MPTCP is analyzed by considering 5G mmWave network and video traffic.

III. PROPOSED S-MPTCP ARCHITECTURE

This section provides an outline of our proposed S-MPTCP based system architecture. Here we initially present an overview of the overall system architecture which is then followed by elaborating the working of each module. The depiction of overall system architecture is given in Figure 1. It shows that the system is composed of two main modules (a) The Analytics Layer & (b) The Control Panel Layer. The Analytics Layer, which is also called as the upper layer and resides on the Hardware Abstraction Layer (HAL) serves for the analysis, classification and providing feedback to the lower layer. It is further subdivided into two modules i.e. (i) the Classifier module and (ii) the Feeder module. Both modules acquire the cross-layer feedback and useful information and disseminate it to the lower layer (Control plane layer). The Control Plane Layer has the actual controller which manages and controls the MPTCP sub-flows. It utilizes the obtained feedback from Analytics layer to set or tune the MPTCP sub-flows control according to the information.

Now, each layer and the related modules will be discussed in detail.

A. ANALYTICS LAYER

As mentioned earlier, the analytics layer is composed of two modules, among which the first one is Packet Classifier module. This module classifies the socket and the type of connection. The second module is the Feeder module which is used to analyze the user equipment (UE) functionalities and determine the efficient use of MPTCP to obtain enhanced throughput with optimum data and power usage.

The packet classifier appears in three different versions as shown in Figure 1, (i) User input-based classifier, (ii) Operator based classifier and (iii) Auto classifier module. The discussion of each classifier module version is as follows.

(i) User input-based classifier: This classifier version gives preference to the user and makes classifications on the basis of user input. Our designed user space application is shown in Figure 2. It works in four different operational modes.
- *Hassle Free Mode*: This mode allows the users to use MPTCP for obtaining better throughput. It is done by employing multiple interfaces simultaneously i.e. both LTE and WiFi.
- *Reliability Mode*: This mode is useful for specific real time network applications such as voice over internet protocols, video conferencing, online gaming, etc., for reliability enhancement. The Connection analyzer module retains the primary interface and switches to

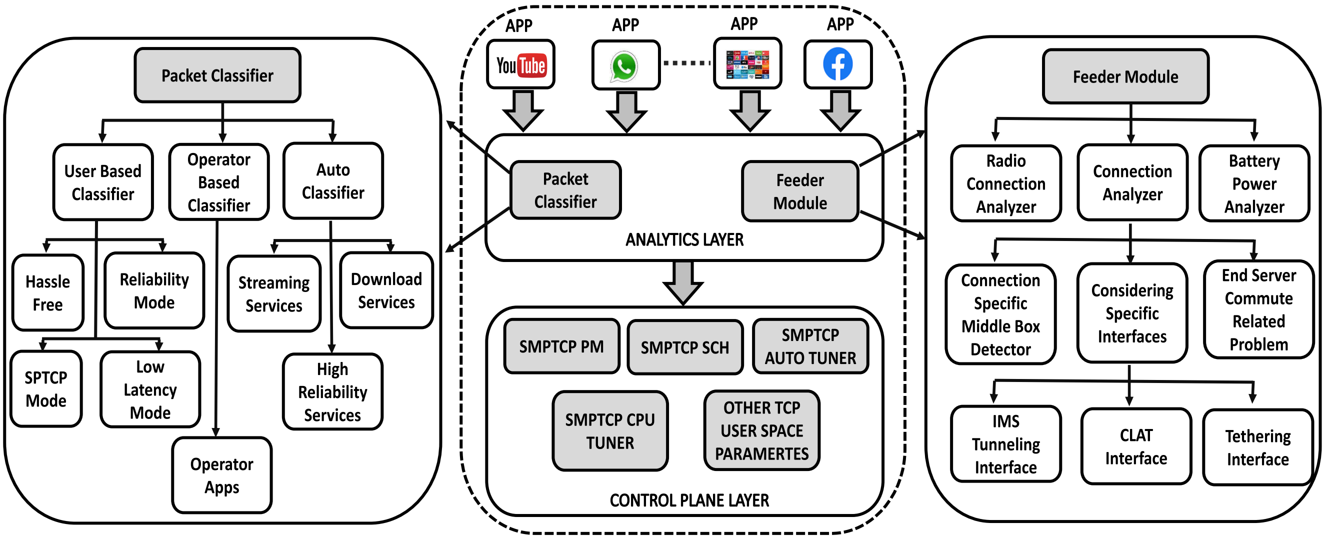

FIGURE 1. S-MPTCP Architecture.

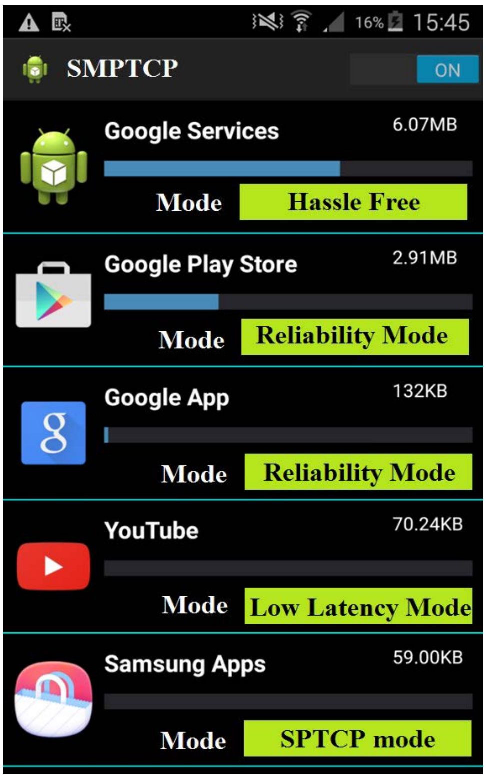

FIGURE 2. User based application.

the mobile interface only when some problem prevails for a considerable duration.

- *Low Latency Mode*: This is a peculiar mode for low latency applications. A redundant mode MPTCP data transmission is assimilated to obtain the minimum possible latency. It is suitable for 5G specific applications having ultra-low latency such as virtual reality, augmented reality and tactile internet applications. Since, latency is the focused criteria in this case, throughput can be compromised to minimum level.

- *Single Path TCP (SPTCP) Mode*: This mode is acquired when the user intents to use the default interface only and to avert the redundant data usage for the least essential applications.

(ii) Operator based classifier: Overloading at user end is a serious concern and to tackle it we propose a proficient mode of interaction with the operator specific applications. Only for few applications the operators don't charge extra to the users. For instance, an Indian operator JIO does not charge extra amount to its users for some of its applications such as JIO TV. Similarly, some Korean and European operators offer special data tariffs for particular applications such as for Facebook and WhatsApp, apart from the usual data package. LTE data usage for these special applications may not be of significant effect because these are accommodated within special or free tariffs. Moreover, we were intended to identify the applications which may face disconnection issues while online streaming or extended downloading time for explicit applications. Users of such applications are suggested to switch on to the MPTCP with the relevant mode. Operator-based classifiers follow the following steps: a) Request is generated by UE for its user subscription plans. b) Based upon the generated request, operator responds to the UE. c) User identification (UID) of the application specified in step b) is fetched from the white listed applications. d) UID fetched is step c) is transferred to the control plane layer to manage the mode appropriately.

(iii) Auto classifier Module This module enables the auto-classification of the user data obtained from various applications and efficiently manage them

Algorithm 1 Video Module Flow

```
1:  Video starts
2:  Calculate C = WB/VB
3:  if C < 1 then
4:      Start DAS
5:      if BR < Th then
6:          Th = min(2*Th, Size of VIdeo/2)
7:          Establish more subflows
8:          TB = Sigma(AB)
9:          if TB < VB then
10:             Go to step 7
11:         else if then
12:             Go to step 5
13:         end if
14:     else if then
15:         Go to step 17
16:     end if
17: else if Video running then
18:     Go to step 2
19: else if then
20:     Stop
21: end if
```

in accordance. Based upon the available services, this category is sub-classified into four main following classes: - *Streaming Services*: The android media framework and streaming module gives us the information that whether the socket is engaged in streaming services or not. For superior streaming services and efficient usage of data we have devised the algorithm ''On-Demand Mobile Data'' as shown in algorithm 1. The rationale behind the deployed algorithm is straight forward, i.e. to utilize it only when demand occurs. For instance, if WiFi is adequately and independently fulfilling the data requirement, only WiFi will be used otherwise the combination of LTE-WiFi will be employed. Algorithm 1 also illustrates the working mechanism of the video module controller block. Here, WiFi Bandwidth and Video Bandwidth is represented as WB and VB respectively. The estimation of WB is based upon the streaming experience and network parameters like signal to noise ratio (SNR), signal strength, latency, packet losses etc. In the start of video streaming, the device calculates the C using formula:

$$C = \frac{WB}{VB}$$

The ratio C determines if the video streaming is smooth or not. The value of $C < 1$, indicates that the video is running with pauses. This is the situation when WiFi alone fails to fulfill the flawless video streaming condition. This information is transferred to the control plane layer where Dynamic Adaptive sub-flow (DAS) controls the mode of the MPTCP operations accordingly. DAS decides the use of MPTCP sub-flow control within current network situation.

In the above figure for Algorithm 1:
WB = WiFi bandwidth
VB = video bandwidth
LP = loaded pivot
CP = current pivot
BR = Boost factor = LP − CP
TB = Total Bandwidth
Th = threshold (initialized to 10 sec)
CB = Current Bandwidth after establishing sub-flow
Sigma (CB) = summation of all CBs
We find the boost factor (BR) by calculating difference between the Load Pivot (LP) and Current Pivot (CP) parameters. Boost factor helps in the identification of video frames loaded in advance. The threshold value (Th) is used to determine whether the multiple interfaces needs to be enabled or not. Selecting a lower threshold value results in a continuous toggling of the interface that probably cost in higher power consumption. Conversely, selecting a higher threshold value may result in a permanently connected running interface that consistently consumes mobile data through the interface. By selecting Th value as $10s$, we obtain the better QoE from our experiment. If *BR* leads the Th, it implies an ample availability of frames to proceed, hence, the inclusion of additional flows is undesired in this situation, otherwise it would have. The proposed algorithm ensures to reduction in LTE data usage by enabling it only when requested. This on-demand utilization of LTE interface when needed saves the power up-to 20%. In existing network system, the start of MPTCP service also initializes the mobile and WiFi interfaces and keep them running continuously.

- *High Reliability Services*: This mode of operation is well suited for applications that do not deem high bandwidth; instead, reliability and latency from services are their major concerns. Applications that are critical for this category may include MCPTT signaling (Mission Critical Push to Talk) and IMS signaling, while the Remote Voice Assistant is classified as high reliability application. In existing network circumstances, the S-MPTCP PM avails the backup or redundant mode for services like these.

- *Download Services*. All android platform-based services use the android download manager for downloading applications. Moreover, there are applications which have their own HTTP stack and data handling procedure such as WhatsApp video downloader, Kakao Talk image downloader, etc. We get into the LIBC layer to assimilate the data flow through it and to classify the services into this category. Our proposed *download algorithm* works on the following procedure:
a) First it examines that if the download is HTTP and subsequently attempts to fetch the download file size through GET request.
b) The download size and current WiFi download status, the approximate completion time is calculated.

Algorithm 2 Download Services

Input: HTTP & NON-HTTP
Output: SPTCP, SMPTCP & MPTCP
if HTTP **then**
  if Object Size $< \alpha_1$ MB **then**
    SPTCP
  else if Object Size $> \alpha_1$ MB **then**
    SMPTCP
  end if
else if NON-HTTP **then**
  if Time to download $< \alpha_2$ Sec **then**
    SPTCP
  else if Time to download $> \alpha_2$ Sec **then**
    MPTCP
  end if
end if

c) If the approximated completion time appears less than the threshold value α_1, we use SPTCP, else other modes of MPTCP will be used.

d) If the file to be downloaded is non-HTTP, or if its size is non-fetchable, we inquire to check the download time by comparing it with a running clock. If the clock moves ahead of the threshold value α_2, we try to switch the MPTCP in some other suitable operational mode by loading the information to the control plane layer specifically for this service. In our experiment we have set α_1 as 20 MB and α_2 as 30 sec [22]. Irrespective of the different network scenarios the α_1 is selected as 20 MB to differentiate download service from the web surfing or web browsing service. This will enable us to choose S-MPTCP only for download service and thereby save the power consumption in smartphones. For thin stream connections the use of S-MPTCP might not benefit the end users. Hence, we select service which continuously exchanges data for more than 30 seconds (α_2) to leverage the benefits of S-MPTCP.

Algorithm 2 provides the run down.

1) FEEDER MODULE

The analyzer module of UE communicates with several modules to fetch the information and delivers the feedback to the control plane layer. The feeder module is depicted in Figure 1. The sub-modules of the feeder module listen the various UE states. Explanation of feeder module's sub-modules is given below.

1) Radio Conditions Analyzer: This module is responsible to comprehend and evaluate the operational parameters of radio. The network radio performance score is calculated through WiFi related parameters like the Link Bandwidth, Received Signal Strength Indicator, Estimated Bandwidth, etc. However, on the contrary, the score of mobile network performance is obtained from the parameters such as Signal to Noise Ratio, Received Signal Strength Indicator and Reference Signal Received Quality. The scores obtained from these parameters helps in deriving relative weightage for paths which are then used in various classifier modules. Beside score estimation and paths weightage assignment, it carries some more useful mobile information like Radio Access Technology change, Data Limit Warning, Data Limit Exceeded, etc. Algorithm 3 depicts the whole process in the form of flow chart.

Algorithm 3 Radio Condition Analyzer

Input: RAT Type
Output: Radio related operational parameters
On RAT change identity, RAT type
if RAT type set with MPTCP **then**
  Use full mash mode
else
  Use backup mode
  MPTCP Kernel
end if

2) Battery/Power Analyzer: The idle time power drainage has been a serious concern associated to MPTCP based devices. Usually, MPTCP devices experience around 30% higher power consumption in contrast to the non-MPTCP devices. It is also observed that the Keep-alive feature and Dead Peer Detection (DPD) method-based application generates multiple spikes that keeps awake the Application Processor. The power analyzer module monitors the actual current power status of the device i.e. during idle state, doze condition, screen off state, deep sleep condition and also communicates it to the connectivity services and path manager. This module also keeps track the usual sleep cycle of smartphones, which is used by the S-MPTCP PM to efficiently utilize the sub-flows for power saving purpose. In case of android sets, SSRM is the module that monitors and intimates the battery conditions. This module can further obtain the information of battery health, battery charging level, battery temperature, etc. and sends this information in the form of feedback to the control plane layer to recover from the battery related issues. The process followed by this module is depicted in Algorithm 4.

3) Connection Analyzer: Nowadays, the majority of smartphones are deployed with socks proxy-based approach [15]. The connection analyzer module is responsible to fetch connection specific parameters information as explained below. a) *Connection Specific Middle-box detectors*: We try to determine the presence of middlebox, specially in WiFi or LTE path. The presence of middlebox limits the use of MPTCP options. This situation restricts the use of MPTCP and the system returns to SPTCP. Hence, we ignore these kinds of connections to avoid problems related to redundant interface connection setup which are prone to over drain the battery. b) *End server commute related problem*: Despite being the most popular and widely used

Algorithm 4 Battery Power Analyzer Flow Chart

1: Input: Stop Battery Heat Level
2: **if** MPTCP Running **then**
3:     **if** Level Stop < Stop threshold **then**
4:         Go to step 10
5:     **else**
6:         Control subflows
7:         Go to step 11
8:     **end if**
9: **else**
10:     Scale down CPU frequency
11:     MPTCP MP path manager, MPTCP Kernel
12: **end if**

approach in smart phones, socks proxy-based approach offers some common problems to few applications. For instance, in case of socks-based proxy or using any other proxy in between, banking applications may always pose errors. Because of such problems, we try to restrict such connections and intimate the control plane to act appropriately. c) *Consideration special interfaces*: Some special interfaces pose several problems as discussed below: - IMS and tunneling interfaces: Dedicated IMS interface is equipped in most of the devices now a days. When devices attempt to establish the connection, full mesh connection is established by default and the MPTCP path manager seeks to establish the sub-flows for all these connections. We try to recognize these interfaces and notify the control plane layer to act appropriately. - CLAT (Communication Line Adapter for Teletype) interface: XLATING interfaces demand an exceptional attention while using MPTCP, otherwise it may cause cross connections; which may deteriorate the throughput and fairness of the connections. Consequently, we tend to ignore the CLAT interface and direct the S-MPTCP PM to create only one to one connection such as IPv4 WiFi interface connection with IPv4 server and IPv6 mobile interface connection with IPv6 server etc. - Tethering interface: Tethering connections are local interfaces to transfer the packets among connected devices. For example, the tethering of mobile hotspot with mobile backhaul; and similarly, the WiFi sharing connection with WiFi backhaul. These connections demand additional sub-flows usage at the back end that may affect the fairness of connections. We cater the demand of android tethering module by communicating it to the S-MPTCP PM.

B. CONTROL PLANE LAYER

The control plane layer comprises of 1) The S-MPTCP PM (Path Manager), 2) The S-MPTCP SCH (Scheduler) and 3) The S-MPTCP Auto Tuner Module.

1) The S-MPTCP PM: The establishment of sub-flows is managed by the S-MPTCP PM module. These may include Sub-flow creation, Sub-flow removal, and etc. Similarly, the mode of operations such as FULLMESH, BACKUP, SPTCP mode, etc. is also managed by the S-MPTCP PM.
2) The S-MPTCP SCH: The MPTCP Schedulers play a vital role in scheduling the packets amongst the sub-flow. The scheduling decision is made on the sender side and can follow various approaches based on the need. The default MPTCP provides the following schedulers:
   2A. Round Robin: The packets are scheduled in Round Robin fashion amongst the available subflows.
   2B. LowRTT: In this scheduler, the packet is pushed in the lowest RTT path until the congestion window is filled. If the congestion window of the lowest RTT path is filled, then the second-lowest RTT is preferred, and so on.
   2C. Re-transmission Penalization (RP): The head-of-line (HOL) blocking can be caused if the packets are pushed in the same subflow. This design proposed opportunistic scheduling by pushing the re-transmitted packet in the available subflow. It can utilize the subflow effectively and also reduce the jitters and delay.
   2D. Buffer-bloat Mitigation (BM): If RP is used, it might fill the high BDP path since it has large buffers. Hence, BM is designed by restricting the data sent on each subflow. Therefore, it avoids buffer bloating by monitoring the minimum Smoothened Round-trip-time.
   2E. Redundant Scheduler: This scheduler sends the data redundantly on all the available paths.
   2F. Backup Scheduler: The backup scheduler pushes the data in the primary path. If the priority of path changes in the path manager, it would update the primary interface and push the data on the new primary path. The previous primary path is set as backup path. If the primary path undergoes a failure, the data packets are pushed in the backup path.
   In our experiment, we have used LowRTT scheduler and Backup Scheduler based on the MPTCP PM indication. The LowRTT scheduler is the default scheduler in MPTCP open source [24]. SMPTCP SCH module is responsible to schedule the data based upon the received information from the analytics layer.
3) The S-MPTCP Auto Tuner Module: This module has the capability of configuring different connection parameters not only system wide but also on the per connection basis. The configuration is done on the basis of the feedback from the analytics layer. Discussion on some prominent parameters is given below:
   - S-MPTCP Buffer Tuner: S-MPTCP Buffer Tuner: In conventional MPTCP, by default, a common TCP buffer value is assigned to all connected interfaces. In order to improve the system-wide throughput or per-connection throughput we efficiently tune the buffer values on the basis of feedback from analytics layer.

Let B_i represent the buffer size to be set for the particular interface i. The value of the buffer size is decided based on the type of network associated and as per Android default. For example, if LTE cellular network is connected then the B_{LTE} is 8 MB. Similarly, if 2.4 GHz Wi-Fi network is connected then the $B_{WiFi2.4}$ is 4 MB. When we use "n" interfaces in our SMPTCP then the buffer to be set is the cumulative buffer value and is upper bound to max_buff (12 MB). This can be written as:

$$\text{final\_buff} = min(\sum_{1}^{n} B_i, \quad \text{max\_buff})$$

where the final_buff is the TCP buffer set while using S-MPTCP solution.

- SMPTCP CPU Tuner: The information received regarding the WiFi and LTE path characteristics (e.g. connections type, RTT difference, maximum TP) determines the CPU related parameters to set. On the basis of these information, to obtain better throughput, we fine tune the CPU related parameters, such as the CPU affinity, CPU governor, CPU clock frequency, per CPU processing, etc.
- Other TCP UserSpace Parameters: In addition to the afore mentioned parameters, we consider to tune some more parameters on the basis of the obtained feedback.
  The parameter includes: a) *tcp_no_metrics_save* (default 1): This is set as 0 so that the TCP starts in the slow start phase during every experiment
  b) *tcp_congestion_algo* (default: olia for MPTCP): We use BBR in our experiment as BBR [23] performs faster than the coupled congestion control algorithms
  c) *tcp_low_latency* (default 0): Set as 1 in our experiment to disable the nagling algorithm.
  d) *interfaceMTU*: If multiple interfaces are connected in the SMPTCP, the MTU value is set as the minimum among the connected interface. Setting different MTU sizes can cause fragmentation on one of the subflows, which degrades the throughput. Therefore, we set the minimum value to avoid the fragmentation issue.

IV. EXPERIMENTAL RESULTS

The assessment of live air experimental setup is discussed here and then the experimental results will be evaluated.

A. LIVE AIR EXPERIMENTAL SETUP

We are considering baseline MPTCP available in [24]. Most of the implementations of MPTCP requires change in both the client as well as server side. Our proposed solution deals with only client side. We did modifications only on client side. For our implementation we do not require server side changes. We did not find any previous research work evaluating ONLY client side modifications and testing it on live air. Hence, we only compare our proposed SMPTCP with baseline MPTCP which is optimized for open source version with various contributors. To evaluate the proposed methodology, we performed the live air experiments in our lab of Samsung Head Quarters, South Korea. The gadgets we used for the experiment was Samsung Galaxy Note 20 and Galaxy S21. The telecom operators we choose for the experiments have either tested or deployed the MPTCP successfully i.e. KT/SKT (MTU 1440/1450 Bytes) of South Korea, and Jio of India (MTU 1500 Bytes). The four types of WiFi we used to evaluate the S-MPTCP were having an aggregate speed of 1300 Mbps, 600 Mbps, 289 Mbps and 54 Mbps. Similarly, we also used the LTE band 8 (10 MHz), LTE band 3 (20 MHz) and LTE band 1 (10 MHz) in non-Carrier Aggregation (CA) mode, 2 (CA) mode and 3 (CA) mode in our experiments. The average speed that CA LTE offers is in the range of 70 to 350 Mbps while the non-CA LTE gives a maximum speed of 70 Mbps. Our complete experimental setup is given in Figure 3 and the parameters used in live air experiment is

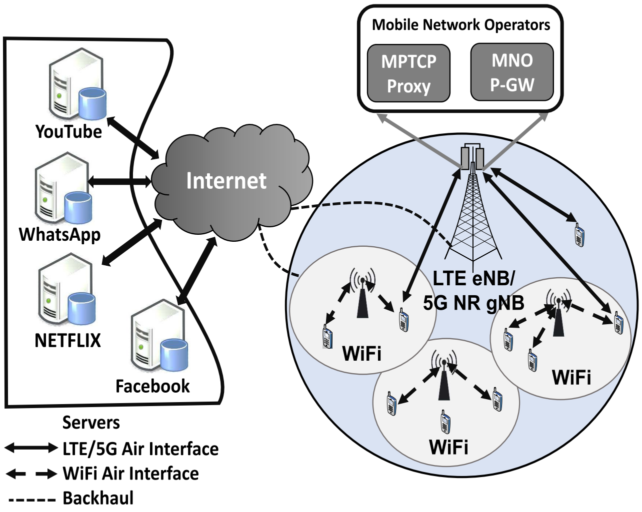

FIGURE 3. Experimental Setup.

TABLE 2. Experiment Parameters.

Parameter	Value
SKT/KT (Korea)-LTE & 5G	MTU: 1450/1440 Bytes
Jio (India)-Only LTE	MTU: 1500 Bytes
WiFi	54 Mbps, 289 Mbps, 600 Mbps & 1300 Mbps
LTE: Band [1, 2, 8]	[10, 20, 10] MHz Non-CA => 70 Mbps, CA => 70 ~ 300 Mbps
5G BW	Farther from gNB: 300 Mbps to 500 Mbps Closer to gNB: 800 Mbps to 1 Gbps
Video characteristics:	Rest: 1920×1080 FPS: 60 Buffer Health: 80 ~ 110 s

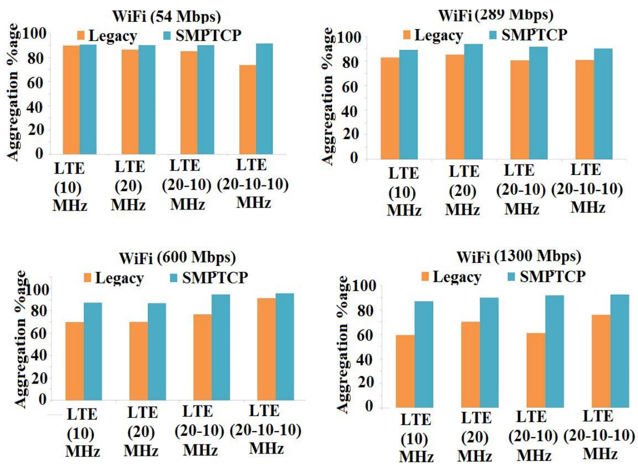

FIGURE 4. Aggregation percentage comparison for different combinations of WiFi and LTE.

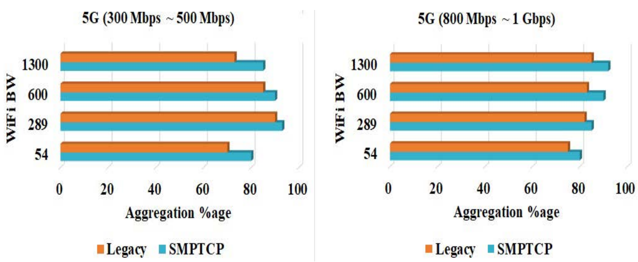

FIGURE 5. Aggregation percentage comparison for different combinations of WiFi and 5G.

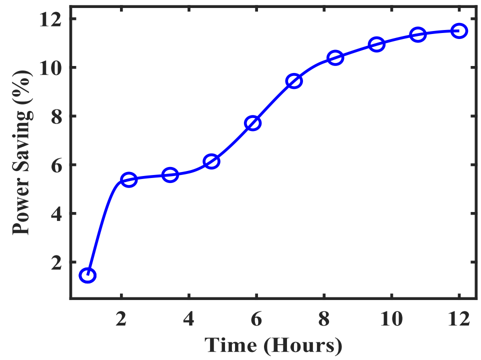

FIGURE 6. Power Savings.

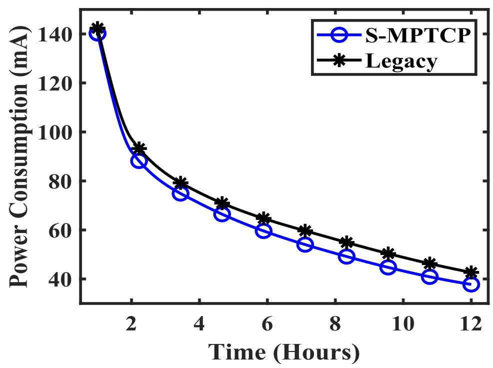

FIGURE 7. Power Consumption Comparison.

enlisted in table 2. To compare SMPTCP with legacy MPTCP we have used the setup as delineated in [24].

B. S-MPTCP EVALUATION

To evaluate the effectiveness of S-MPTCP we measure the aggregation benefit by utilizing available interfaces. The aggregation percentage is given as:

$$Aggregation\ \ Percentage = \frac{TP_{SMPTCP}}{\sum_{i=1}^{n} TP_i} \quad (1)$$

where TP_{SMPTCP} is throughput of S-MPTCP, n is the total number of used interface, and TP_i is the throughput of specific interface i used in the S-MPTCP.

The comparison of aggregation percentage we obtained against each combination of WiFi & LTE (single carrier, non-CA & CA) and WiFi & 5G (farther from gNB and nearer to gNB) is given in Figure 4 and Figure 5, respectively. From Figure 4, We clearly see that the S-MPTCP ensures a higher aggregation percentage with respect to the legacy MPTCP. The auto tuning mechanism of control layer on the basis of detailed feedback provided by the analytics layer gave us a maximum aggregation percentage gain of 50.39%. Consequently, a higher throughput was obtained in case of our proposed S-MPTCP scheme. It is worthwhile to mention that in any case the throughput obtained from S-MPTCP will never fell below the throughput value of SPTCP. Whereas, the SPTCP outperforms the MPTCP in conditions when there exists higher difference in RTT of WiFi and LTE. Figure 5 presents the aggregate percentage comparison for a different combination of WiFi and 5G bandwidth. As observed from figure 5 the aggregate bandwidth utilization with legacy is low as compared to 5G. For WiFi bandwidth 289 and 5G bandwidth 300 Mbps ∼ 500 Mbps the aggregate percentage of 5G NR is almost 90% and aggregate percentage of legacy is almost 85%. It is noticed from figure 5 for WiFi bandwidth 1300 and 5G bandwidth 800 Mbps ∼ 1 Gbps the aggregate percentage of 5G NR is more than 90%.

Another crucial benchmark of smart-phone industry is the power saving. Hence, we examined and juxtaposed the power saving and power consumption of conventional MPTCP and S-MPTCP. The power saving percentages given in the graph of Figure 6 is when the screen of the mobile phone is in OFF state for a total duration of 12 hours. We found that through S-MPTCP, we achieved a maximum power gain of around 11.5% in contrast to the legacy MPTCP. The comparison of power consumption statistics in mA between legacy MPTCP and S-MPTCP is shown in Figure 7. In this case also, after a continuous use of 12 hours, the S-MPTCP outclassed the conventional MPTCP by consuming overall 13% less power compared to the legacy MPTCP. This was made possible by our proposed MPTCP controller system that created an efficient inter-layer coordination between the smart analytics layer and control plane layer.

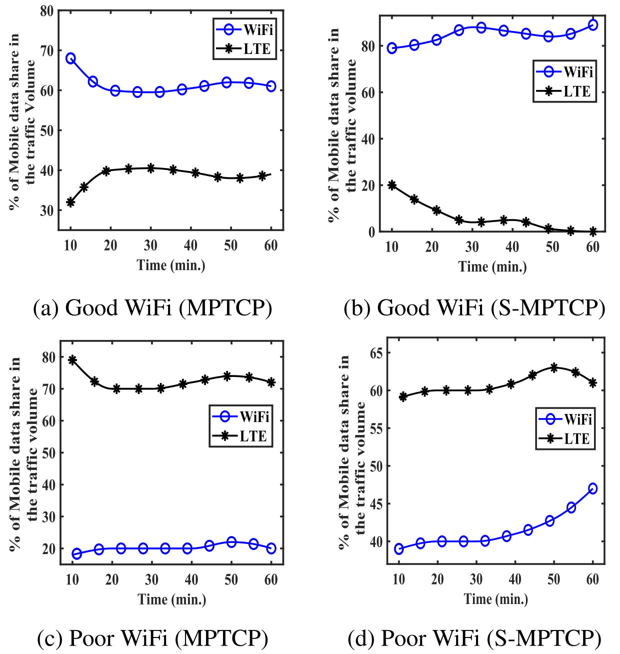

FIGURE 8. Comparison of LTE data consumed.

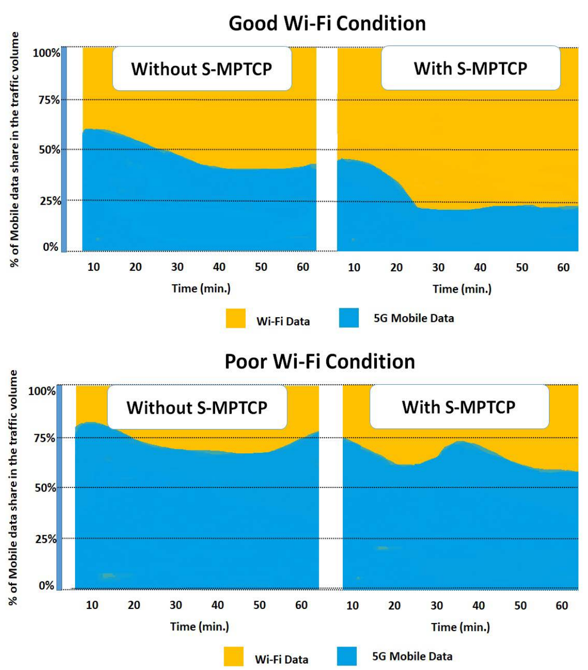

FIGURE 9. Comparison of 5G NR data consumed.

In contrast to the conventional MPTCP, our suggested system can efficiently determine the time and magnitude of cellular data to be used over mobile network sub-flows. We tested the video streaming from YouTube server for 1 hour using legacy MPTCP and S-MPTCP under good and poor WiFi conditions (1 Mbps and 0.5 Mbps respectively). We found that for all WiFi conditions, S-MPTCP saves more cellular data compared to the legacy MPTCP as shown in Figure 8. In case of conventional MPTCP, 1440 MB and 720 MB of data was consumed for good and poor WiFi conditions respectively. However, for S-MPTCP, 2520 MB and 2160 MB of the WiFi data was consumed on both good and poor WiFi conditions respectively. Similarly, in all WiFi conditions, more LTE data was consumed by the legacy MPTCP as compared to that of S-MPTCP.

Figure 9 presents the compared the 5G NR data consumption with and without S-MPTCP in case of Good WiFi and Poor WiFi condition. The share of 5G mobile data With S-MPTCP for both Good and poor WiFi is low as compared to Without S-MPTCP. For good WiFi, the average utilization of 5G data is 27% whereas in the case of poor WiFi the average utilization of 5G data is more than 60%. It is observed from Figure 9 that initially there will be a peak in 5G data, where 5G and WiFi loads the buffer together. After that 5G will be less used as the advance buffer is loaded mostly from the WiFi itself.

V. CONCLUSION

We present here a novel smart MPTCP controller concept for Next Generation Mobile Networks. In order to combat the challenges faced by legacy MPTCP, our proposed system efficiently manages and controls its sub-flows on the basis of path characteristics and interface. The architecture of our proposed system is mainly composed of two layers i.e. the Analytics Layer and Control Layer. The analytics layer acquires the cross-layer feedback and useful information about the connections and provides them to the control layer which is the actual controller of our system and responsible for the efficient management and control of MPTCP sub-flows. The proposed scheme was also verified through live air experiments and simulations at Samsung Electronics, Head Quarters Lab. Both simulation and experimental results confirm that our approach can efficiently manage and control the sub-flows to provide higher throughput at the cost of minimum connection overhead. We achieved an overall 13% power gain and 50% LTE mobile data saving compared to the conventional MPTCP solutions. For 5G NR and good WiFi conditions, the average utilization of 5G Data with S-MPTCP is 27%, where as it is 53% for legacy MPTCP solutions. These promising results validates the competency of our proposed system.

REFERENCES

[1] T. Barnett, S. Jain, U. Andra, and T. Khurana, "Cisco visual networking index (VNI), complete forecast update, 2017–2022," Americas/EMEAR Cisco Knowledge Network (CKN) Presentation, Cisco Systems, San Jose, CA, USA, White Paper, 2018.

[2] Y. Xing, J. Han, K. Xue, J. Liu, M. Pan, and P. Hong, "MPTCP meets big data: Customizing transmission strategy for various data flows," *IEEE Netw.*, vol. 34, no. 4, pp. 35–41, Jul. 2020.

[3] A. Ford, C. Raicu, M. Handley, S. Barre, and J. Iyengar, *Architectural Guidelines for Multipath TCP Development*, document RFC 6182, Mar. 2011.

[4] B. Y. L. Kimura, D. C. S. F. Lima, and A. A. F. Loureiro, "Packet scheduling in multipath TCP: Fundamentals, lessons, and opportunities," *IEEE Syst. J.*, vol. 15, no. 1, pp. 1445–1457, Mar. 2021.

[5] M. R. Kanagarathinam, S. Singh, V. S. Kumar, and K. J. Moon, "S-MPTCP: A smart multipath TCP controller for next generation mobile network," in *Proc. IEEE 20th Int. Conf. High Perform. Comput. Commun., IEEE 16th Int. Conf. Smart City, IEEE 4th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Jun. 2018, pp. 1165–1170.
[6] H. Tong, T. Wang, Y. Zhu, X. Liu, S. Wang, and C. Yin, "Mobility-aware seamless handover with MPTCP in software-defined HetNets," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 1, pp. 498–510, Mar. 2021.
[7] Y. Zhang, J. Tourrilhes, Z.-L. Zhang, and P. Sharma, "Improving SD-WAN resilience: From vertical handoff to WAN-aware MPTCP," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 1, pp. 347–361, Mar. 2021.
[8] T. Zhang, S. Zhao, and B. Cheng, "Multipath routing and MPTCP-based data delivery over manets," *IEEE Access*, vol. 8, pp. 32652–32673, 2020.
[9] I. Gojmerac, T. Ziegler, F. Ricciato, and P. Reichl, "Adaptive multi-path routing for dynamic traffic engineering," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, San Francisco, CA, USA, 2003, pp. 3058–3062.
[10] M. Coudron, S. Secci, G. Pujolle, P. Raad, and P. Gallard, "Cross-layer cooperation to boost multipath TCP performance in cloud networks," in *Proc. IEEE 2nd Int. Conf. Cloud Netw. (CloudNet)*, San Francisco, CA, USA, 2013, pp. 58–66.
[11] L. Li, N. Hu, K. Liu, B. Fu, M. Chen, and L. Zhang, "AMTCP: An adaptive multi-path transmission control protocol," in *Proc. 12th ACM Int. Conf. Comput. Frontiers*, Ischia, Italy, May 2015, pp. 1–8.
[12] M. Sandri, A. Silva, A. Lucio Rocha, and L. Fabio Verdi, "On the benefits of using multipath TCP and OpenFlow in shared bottlenecks," *Adv. Inf. Netw. Appl. (AINA)*, Gwangju, South Korea, 2015, pp. 9–16.
[13] H. Nam, D. Calin, and H. Schulzrinne, "Towards dynamic MPTCP path control using SDN," in *Proc. IEEE NetSoft Conf. Workshops (NetSoft)*, 2016, pp. 286–294, doi: 10.1109/NETSOFT.2016.7502424.
[14] S. Zannettou, M. Sirivianos, and F. Papadopoulos, "Exploiting path diversity in datacenters using MPTCP-aware SDN," 2015, *arXiv:1511.09295*.
[15] M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, and L. Jones, *SOCKS Protocol Version 5*, document RFC 1928, 1996.
[16] J. Han, Y. Xing, K. Xue, D. S. L. Wei, G. Xue, and P. Hong, "Leveraging coupled BBR and adaptive packet scheduling to boost MPTCP," 2020, *arXiv:2002.06284*.
[17] P. Ignaciuk and M. Morawski, "Discrete-time sliding-mode controllers for MPTCP networks," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 10, pp. 6029–6039, Oct. 2021.
[18] Y. Luo, "A real-time video streaming system over IPv6+MPTCP technology," in *Proc. Workshops Int. Conf. Adv. Inf. Netw. Appl.* Cham, Switzerland: Springer, 2019, pp. 1–4.
[19] A. Elgabli and V. Aggarwal, "SmartStreamer: Preference-aware multipath video streaming over MPTCP," *IEEE Trans. Veh. Technol.* vol. 68, no. 7, pp. 6975–6984, Jul. 2019.
[20] S. Nagayama, D. Cavendish, D. Nobayashi, and T. Ikenaga, "Path switching schedulers for MPTCP streaming video," in *Proc. IEEE Pacific Rim Conf. Commun., Comput. Signal Process. (PACRIM)*, Aug. 2019, pp. 1–6.
[21] J. Hwang, A. Walid, and J. Yoo, "Fast coupled retransmission for multipath TCP in data center networks," *IEEE Syst. J.*, vol. 12, no. 1, pp. 1056–1059, Mar. 2018.
[22] *Webpage Size—Why is it Important? and How do You Optimize It?*. Accessed: Jul. 30, 2022. [Online]. Available: https://tinyurl.com/mtcwzusp
[23] J. Han, K. Xue, Y. Xing, P. Hong, and D. S. Wei, "Measurement and redesign of BBR-based MPTCP," in *Proc. ACM SIGCOMM Conf. Posters Demos*, Aug. 2019, pp. 75–77.
[24] C. Paasch, G. Detal, S. Barré, F. Duchêne, and O. Bonaventure, *Multipath TCP in the Linux Kernel*. Accessed: Mar. 12, 2022. [Online]. Available: https://www.multipath-tcp.org

SUKHDEEP SINGH (Member, IEEE) received the Ph.D. degree from Sungkyunkwan University, South Korea. He is currently a Senior Chief Engineer with Samsung Research and Development India-Bangalore. He has published in IEEE, Wiley, Taylor & Francis, and IET. His research interest includes 5G/6G wireless communications.

MUKESH KUMAR MAHESHWARI received the master's degree from the University of Leicester, U.K., and the Ph.D. degree from Sungkyunkwan University, South Korea. He is currently a Faculty Member with the Department of Electrical Engineering, Bahria University, Pakistan. He has published in IEEE, Wiley, Taylor & Francis, and IET. His research interests include 5G wireless communications and energy-efficient networks.

MADHAN RAJ KANAGARATHINAM (Senior Member, IEEE) a received the B.E. degree in computer science and engineering from Anna University, Chennai, India, in 2012. He has seven years of working experience in design and development of TCP/IP protocols, multipath TCP and UNIX flavored operating systems. He was an Engineer at Aricent Technology (India) Pvt. Ltd. Currently, he is a Chief Engineer with Samsung Research and Development Institute India Bangalore. He is the author of ten articles, and more than 20 inventions. His current research interests include communication and networks which include Pre-6G/Beyond 5G, next generation mobile networks, software defined network architecture, 40 transport layer protocols, and cross layer optimization technique. He is a member of the ACM.

AURANGZEB RASHID MASUD received the master's degree from NED University, Pakistan, and the Ph.D. degree from SKKU, South Korea. He is currently a Faculty Member with the Department of Electrical Engineering, Bahria University, Pakistan. His research interests include display devices and materials.

IQBAL QASIM, photograph and biography not available at the time of publication.