IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received October 29, 2020, accepted November 15, 2020, date of publication November 27, 2020,
date of current version December 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3041106

A Session-Based Cross-Layer Approach for Seamless Connectivity in Next-Generation Mobile Networks

KARTHIKEYAN ARUNACHALAM[1], (Senior Member, IEEE),
SHIVA SOUHITH GANTHA[1], (Member, IEEE),
SWETA JAISWAL[1], (Member, IEEE),
JAMSHEED MANJA PPALLAN[1], (Member, IEEE),
SEONGKYU SONG[2], (Member, IEEE),
AND ANSHUMAN NIGAM[1] (Member, IEEE)
[1]NextGen Networks, Samsung R&D Institute India-Bangalore, Bengaluru 560037, India
[2]Advanced Communications Research, Samsung Seoul R&D Campus, Seoul 06765, South Korea

Corresponding author: Karthikeyan Arunachalam (karthikeya.a@samsung.com)

ABSTRACT The Fifth Generation (5G) network provides a platform for emerging technologies such as vehicular communication, massive Internet of Things (IoT), and tactile internet. It is paving the way to a modern autonomous industry with mobility as a key requirement to achieve seamless end-to-end connectivity. But the current transport layer is not well equipped with the evolving lower layers to deal with the dynamics of the network conditions during mobility. The variation in network conditions and frequent disturbance in lower layers due to handovers causes performance degradation in the transport layer. Hence, we propose a novel solution called ID oriented Socket Layer (IoSL), an end-to-end solution for futuristic applications. It is an ID-based software solution that maintains a unique identifier among the transport layer connections to provide seamless connectivity to the end devices. During user mobility and frequent handover scenarios, it communicates with the lower layers and effectively controls the transport layer for achieving improved Quality-of-Service (QoS) with enhanced user experience. We prototyped IoSL in mobile devices and evaluated the performance using the ns-3 simulator as well as live-air experiments. The experiments are conducted using Wi-Fi and 5G networks, and the results show a significant reduction in reconnection latencies around 33% and improved end-user throughput by up to 16%.

INDEX TERMS Transport protocols, cross-layer design, mobility, TCP/IP.

I. INTRODUCTION

The 5G network is being deployed over multiple countries to provide a platform for various emerging technologies such as Virtual Reality(VR)/Augmented Reality(AR), autonomous cars, massive IoT, and tactile internet. With 5G development, there will be a steep increase in the number of connected devices [1] with diverse network infrastructure, which also means more radio link fluctuations and frequent handovers, as shown in Fig. 1. Upcoming advanced IoV (Internet of Vehicles) applications will be computation and energy consuming. Since the individual vehicles own limited computational resources and have energy constraints, Mobile Edge Computing (MEC) is a promising solution for offloading heavy computation tasks to resource sufficient edge servers connected to 4G/5G macro-cells or Road Side Units (RSUs) [2], [3]. Also, a single type of access technology may not satisfy all the requirements of the vehicular services [4]. In order to provide agile and low latency service, it is vital to improve the end-to-end performance at the transport layer to provide seamless end-to-end connectivity during mobility scenarios and vertical (intra network) handovers.

The TCP/IP is the predominant transport layer protocol suite that governs the current internet. TCP aimed to provide reliable end-to-end connectivity for wired networks. Therefore, any changes in the lower layers will affect the reliability of ongoing TCP connections [5]. Even though 5G systems take care of mobility in lower layers, IP addresses

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

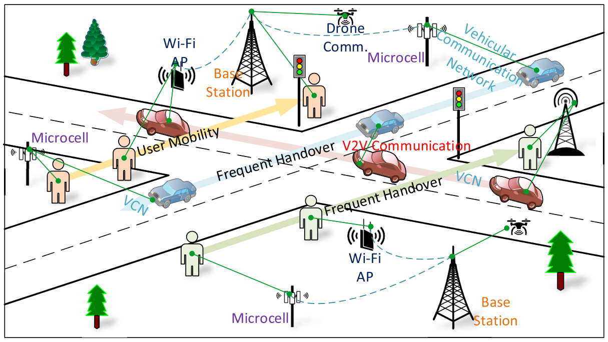


FIGURE 1. Futuristic devices with diverse network infrastructure.

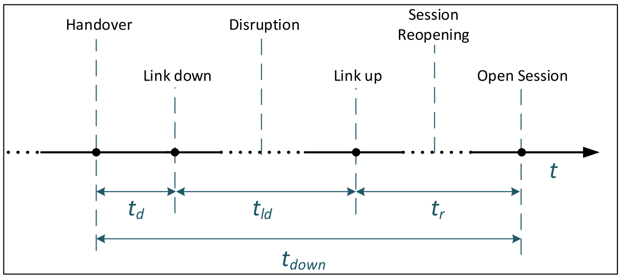


FIGURE 2. Handover events during mobility, which causes frequent link disconnections for any ongoing connection.

can change during a TCP connection, due to inter-network mobility from cellular to a Wi-Fi network or vice versa. This inherent fluctuations in the radio signal conditions, and the handover from micro to the macro cell during user mobility, create performance degradation in the current transport layer. Similarly, the radio signal fluctuations in the lower layers are not notified immediately to the upper layers, which impacts the performance of transport layer significantly. Hence, it is crucial to take care of mobility aspects in the transport layer and enable cross-layer communications for better utilization of resources.

Along with mobility, the heterogeneous wireless network designed for the beyond 5G network aims to have massive coverage and capacity with ubiquitous connectivity [6]. To achieve it, the presence of non-terrestrial network technologies, such as airborne and space-borne, is significant, which also brings new challenges with itself, such as mobility of the base station. Moreover, the performance of next-generation applications such as autonomous driving and tactile internet heavily depends on seamless migration over these heterogeneous networks. Hence, support for fast handover and seamless mobility in transport layer is crucial for future real-time services and applications for enhanced Quality of Service (QoS).

Considering the limitations of transport layer and the importance of mobility requirements, we have designed a cross-layer solution for the transport layer to provide seamless connectivity to the futuristic applications. It is designed and implemented as a transparent middleware solution that is easily deployable as a single-ended (client-side) solution as well as a double-ended (both client and server-side) solution for enhanced end-to-end connectivity. By describing IoSL as a transparent solution, we imply that deployment of IoSL does not require any modifications in the socket programming layer, existing applications and middle boxes.

The remainder of the article is structured as follows. The following, Section II discusses the motivations of this proposal. Section III then presents the prior arts and background study. We describe the design and architecture of the proposed solution on Section IV. Section V explains the crucial design features of the solution and Section VI provides the mathematical model for performance analysis. In Section VII, we evaluate the performance of IoSL using simulated environment. Section VIII presents the experimental results of the solution before concluding this article on Section IX.

II. MOTIVATION

The design of the current transport layer imposes many restrictions to achieve the mobility requirements of futuristic applications. The present mobile devices are equipped with multiple wireless interfaces, which increases the handover scenarios in these diverse network infrastructure [7]. However, the current transport layer does not support IP mobility by default. Also, fluctuations at the radio level affect the performance of transport layer protocols in terms of session continuity and bandwidth utilization. This section further discusses such challenges, trends, and future requirements, which motivated us to pursue our proposal.

A. MOBILITY - A FUTURE NETWORK REQUIREMENT

As we know, the 6G KPI's are set to achieve the high mobility requirement up-to 1000 km/h with mobile base stations for ubiquitous connectivity. It forces us to think, how to overcome the current transport layer limitations to achieve these futuristic requirements. There is a need for well-designed mobility solutions to handle the inter (vertical) and intra-network (horizontal) handover scenarios smoothly especially, to recover the connection state to normalcy as fast as possible during re-establishments. Even a small improvement in the reconnection times during mobility can have immense benefits for delay sensitive IoV applications. The Fig. 2 shows the sequence of events happening during user mobility. It shows the impact of frequent link disconnections [8] and handovers during end-user mobility scenarios. Hence, it is important to design a transport layer solution for faster connection resumption and restore the connection state for vertical and horizontal handover scenarios.

B. LIMITATIONS OF THE CURRENT TRANSPORT LAYER

The current Internet Protocol suite mainly relies on TCP to ensure end-to-end communication along with additional features such as reliability, flow and congestion control. There are many variants of congestion control, such as BIC, CUBIC, BBR, and many more, designed to maximize for better utilization of available bandwidth. Even though these algorithms work perfectly in different network scenarios, but they inaccurately detect the current state of the network and its congestion. They use timeouts and duplicate acknowledgment parameters to detect packet loss [9]. However, these parameters are not enough and lead to unnecessary packet retransmission and data rate reduction. Similarly, other transport layer protocols such as SCTP, MPTCP, and QUIC use similar congestion control algorithm and inherits the same mechanisms such as slow start and congestion avoidance as TCP. Hence, it is critical to resolving these limitations for meeting the necessities of next-generation networks.

C. SHORTCOMINGS OF MMWAVES SPECTRUM

The 5G's mmWaves radio technology is considered as a key enabler for achieving the high data rate with ultra-low latency requirements. However, the short-range characteristics of mmWave spectrum are severely affected in Non-Line of Sight (NLOS) conditions, since it is easily absorbed and blocked by obstacles. It also results in channel fluctuations in link layers, which impacts the bandwidth utilization in the transport layer. There are many simulation studies done to show the impacts on mmWaves when a device moves from LOS to NLOS and vice versa, as mentioned in [10]. The results depict that frequent transitions lead to wastage of the available bandwidth and causes buffer bloat issues, which impacts the performance of overall system. Hence, there is a need for appropriately handling these radio level fluctuations in transport layer for improved performance.

D. UNDER UTILIZATION OF LOWER LAYER RESOURCES

As a consequence of the layered architecture of the Internet protocol suite, the transport and lower layers operate independently and do not team up to support mobility or deal with congestion in the network. The only potential support is the Explicit Congestion Notification (ECN) [11] that provides some hints to the transport protocol to react to congestion. However, ECN does not provide accurate information about the location and gravity of the congestion in the network. In addition to this, the network layer routing information also helps to indicate the link down and up events [12], but it is not the best choice for link-layer predictions. However, this link-layer information cannot be utilized by the transport layer to predict the handovers for providing a seamless user experience during mobility.

E. EASE OF DEPLOYMENT

Deployment of new transport layer solutions is the most restrictive problem in the evolution of the transport layer [13], [14]. Ease of deployment is one of the essential considerations for any transport layer protocol solutions. Apart from TCP, many other transport layer protocols such as MPTCP, SCTP, and QUIC are evolving, as an alternative for TCP. Even though these protocols support the mobility and multipathing requirements, the widespread of middleboxes slow down the deployment of these protocols. Moreover, these protocols require API (Application Programming Interface) for connecting applications with the transport layer. Hence, designing a transparent solution over the legacy protocols with mobility and multipathing would be beneficial for futuristic applications.

Considering the above-explained problems, we have designed an easily deployable mobility solution called ID oriented Socket Layer (IoSL). It communicates with the lower layers to identify user mobility and handover scenarios and effectively controls the transport layer for optimal utilization of available bandwidth. IoSL is implemented successfully in the mobile devices and evaluated using an ns-3 simulator as well as the real-time environment. The experimental results show a reduction in reconnection time up to 33% and improvement in throughput up to 16% during mobility.

Novelty and Contribution

- We design a unique ID based software solution for seamless user mobility, which can run over major transport layer protocols such as TCP, SCTP and MPTCP regardless of its characteristics.
- A new congestion control algorithm that uses cross layer information to evaluate the real time network conditions and adjust the data rate for efficiency during radio level fluctuations is implemented.
- The proposed solution is evaluated in a simulated environment consisting of both WiFi and 5G access networks to depict various handover scenarios. The solution is implemented in Samsung devices to conduct live air experiments. Performance results show that our proposed solution improves the end-user throughput and reduces average reconnection latency significantly.

III. RELATED WORK

Support for IP mobility is widely accepted as a necessary requirement for future wireless networks and use cases [15]. There have been many research works that define a clean slate architecture, such as ID Oriented Networking (ION) [16], MobilityFirst (MF) [17] and Information Centric Networking (ICN) [18], with their prime focus on mobility in heterogeneous wireless technologies. The major drawback of these Non-IP solutions is that they define a completely new protocol stack and need changes in all the network nodes, posing serious deployment issues. Kimura et. al. propose a socket layer [12], called SMSL, for link disruption tolerance for TCP. FreezeTCP [19] is receiver side solution that sends zero receiver window probes when radio link disconnection is predicted. Both, SMSL and FreezeTCP are specific to TCP and do not improve upon TCP's congestion control algorithm, leading to resource underutilization.

There are lot of research ideas and proposals for new transport layer protocols that try to solve the limitations of the present transport layer. QUIC [20], a transport layer protocol, uses unique IDs for connection migration during network handovers. However, it's inefficient TCP-like congestion control algorithm results in high reconnection time during disconnections. On the other hand, X-TCP [21] defines a cross layer congestion control algorithm over mmWaves.

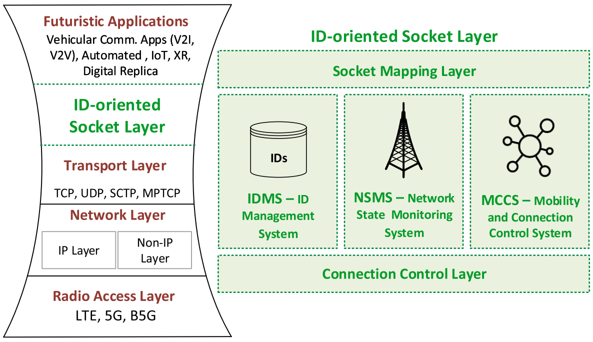


FIGURE 3. Overall Architecture of IoSL solution. IoSL is an ID-based layer between the application and transport layers to guarantee seamless end-to-end communications during mobility.

These solutions try to solve a subset of the transport layer issues and do not specifically consider handover or mobility scenarios.

Considering these limitations and importance of mobility requirement, we have designed an easily deployable software solution for providing seamless connectivity with enhanced QoS for the futuristic applications.

IV. IOSL: DESIGN AND ARCHITECTURE

IoSL is an ID-based solution that uses cross-layer information to analyze the real-time network condition for providing seamless connectivity during vertical and horizontal handover scenarios. As shown in Fig. 3, IoSL defines a new ID-based layer between the application and transport layers for achieving enhanced connectivity to the end devices. It is designed as a transparent middleware solution that maintains unique identifiers for applications and its socket layer connections. During radio link layer fluctuations, IoSL suspends the ongoing transport layer connections and resumes it after the link is recovered. During inter mobility of the network, IoSL re-establish transport layer connections and restores the network state quickly for the fast recovery of the connections.

The IoSL is designed to operate in two different modes, namely double-ended and single-ended modes. In the double-ended mode, it does the peer process communication and assists the transport layer at both ends of the devices for quick recovery of the transport layer connections. In single-ended mode, it acts as a device-only solution and enables cross-layer communication for enhanced transport layer communication. IoSL operates in double-ended mode, by default, to support seamless connectivity during vertical as well as horizontal handovers.

A. IOSL ARCHITECTURE

IoSL consists of three major components, namely, ID Management System (IDMS), Network State Monitoring System (NSMS), and Mobility and Connection Control System (MCCS), as depicted in Fig. 4. The internal architecture of each component is explained further in detail in this section.

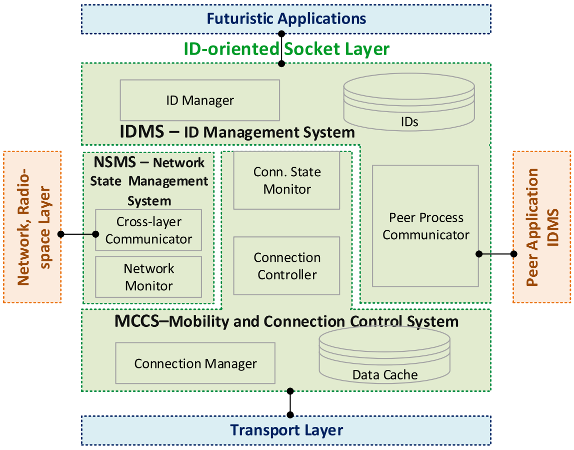


FIGURE 4. Detailed block diagram of IoSL. It uses cross-layer information to analyze the real-time network condition and communicates with peer-process for providing seamless connectivity.

1) ID MANAGEMENT SYSTEM (IDMS)

The prime responsibility of this component is to generate a unique identifier for each transport layer connections. As the name suggests, IDMS is also responsible for managing unique identifiers in an internal database for location-independent communication. These unique identifiers are based on 5-tuple values and connection parameters obtained from cross-layer communication. When application request for a transport layer connection, IDMS allocates and maps the unique identifier to the requested application socket. Also, it deallocates the unique identifier from the application socket when the transport layer connection is closed.

2) NETWORK STATE MONITORING SYSTEM (NSMS)

It monitors the network state in real-time through cross-layer communication with lower layers. It analyzes the radio channel characteristics and its quality parameters to understand the state of the network in lower layers. Also, it analyzes the transport layer parameters such as average RTT, throughput, Retransmission Time Out (RTO), and many more for each transport layer connections. After processing transport and lower-layer parameters, it notifies to MCCS for controlling transport layer connections.

3) MOBILITY AND CONNECTION CONTROL SYSTEM (MCCS)

As the name implies, it controls the transport layer connections when the network conditions vary in the lower layer. Based on the processed information received from the NSMS, MCCS regulates the transport layer parameters such as the congestion window (`cwnd`) and receive window (`rwnd`) for optimal utilization of the available bandwidth. During radio link layer fluctuations, MCCS suspends all ongoing transport connections and resumes them back when the link-layer is recovered. When the ongoing transport layer connection teardown due to inter-network mobility of the device, MCCS

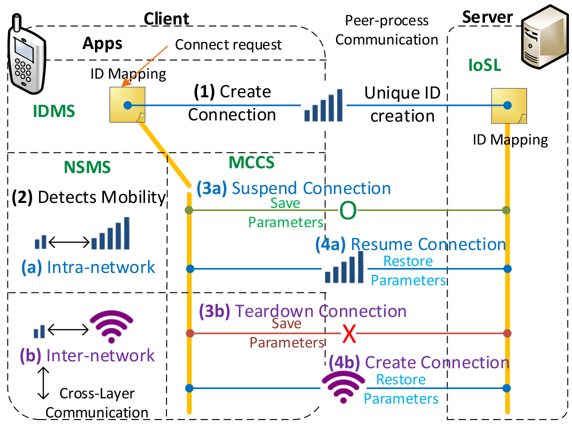


FIGURE 5. ***Double-ended*** **mode of IoSL enables peer to peer communication and assist the transport layer during inter and intra-network mobility for smooth user mobility with fast connection recovery.**

restores the network parameters for the newly migrated transport layer connections. Based on the network conditions, It takes the appropriate decisions for providing seamless connectivity without hampering user experience.

B. OPERATIONAL FLOW OF IoSL

Fig. 5 shows the operation flow of IoSL in double-ended mode. (1) When application request for a transport layer connection, IDMS generates and allocates a unique identifier in its socket management layer for restoring the state of all transport layer connection later. (2) NSMS monitors the radio channel quality through cross-layer communication. It continuously analyzes the channel quality parameters such as CQI (Channel Quality Indicator), SNR/SINR (Signal-to-Interference-plus-Noise Ratio), RSRP (Reference Signal Received Power), RSRQ (Reference Signal Received Quality), and many more, and identifies the (a) inter and (b) intra-network mobility of the end device. (3) MCCS continuously regulates the flow and congestion control of the transport layer during the radio link layer fluctuations. 3(a) Also, it suspends the ongoing transport layer connections when radio link disconnection is predicted. 3(b) Whereas, when inter-mobility is detected while changing to a different network (5G and Wi-Fi), MCCS stores the transport layer parameters and terminates the connections. (4) When the lower layer recovers from the link-layer disconnection, 4(a) MCCS resumes the transport layer connections with restored transport layer parameters. 4(b) In case of a terminated connection (due to inter-network migration), MCCS establishes a new connection with restored transport layer parameters and maps it to the application socket with the help of IDMS. Thus, it transparently recreates the transport layer connection and regulates the data flow of the connection without hampering the user experience.

The operation flow of the single-ended mode, as shown in Fig. 6, is similar to that of double-ended variant, except for peer process communication and support for vertical mobility, since it is installed only at the device side. Hence, single-ended IoSL guarantees the smooth recovery of connections during horizontal handover.

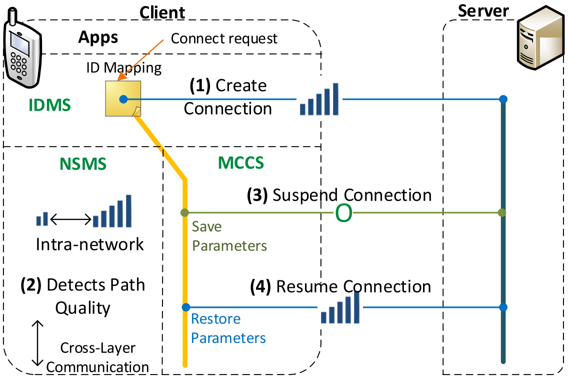


FIGURE 6. ***Single-ended*** **mode of IoSL is a device-side solution that supports seamless connectivity during intra-network mobility.**

V. DESIGN FEATURES OF IOSL

This section explains the design features of IoSL such as easy deployment, cross-layer design, compatibility with major transport layer protocols, and ID layer for mobility, discussed in detail.

A. EASY FALLBACK TO SINGLE-ENDED SOLUTION

IoSL is designed to operate in dual-modes for easy deployment. As described in section IV, IoSL operates in double-ended mode by default and controls transport layer connections at both ends (client and server). In default mode, it continuously communicates with the peer process and exchanges the control parameters, such as `cwnd, rwnd`, with both ends to assist the transport layer for optimal utilization of the available network. If the peer end does not support IoSL, the operation falls back to the single-ended mode and regulates the transport layer connections at the one end of the device.

B. CROSS-LAYER DESIGN OF IoSL

When the application sends data over the transport layer connection, IoSL controls the congestion window based on the radio layer parameters received through cross-layer communication. When the application receives data over the transport layer connection, IoSL modifies the receive window and controls the sender through window advertisement. Whenever a handover or link-layer disconnection is detected, IoSL controls the congestion and flow control parameters for avoiding packet loss and excessive buffering in the queue. Also, IoSL ignores the RTT value calculated during the temporary radio signal fluctuations and optimizes the cumulative RTT for the better utilization of available bandwidth. Thus, IoSL analyzes the radio layer parameters and controls the transport layer for the improved performance in the wireless network.

C. COMPATIBILITY WITH MAJOR TRANSPORT LAYER PROTOCOLS

IoSL is designed to operate independently of the underlying transport layer protocols such as TCP, SCTP, and MPTCP. In particular, all these protocols use similar congestion and flow control mechanisms. SCTP is known for its unique features, such as multi-homing and multi-streaming. SCTP's congestion and flow control mechanisms are designed for association level only and not supported for stream level. So, IoSL generates and maps the unique identifier for only SCTP associations, for resuming and restoring SCTP parameters. MPTCP is known for multi-pathing features, which creates sub-flows over multiple interfaces for path redundancy, and each sub-flows of MPTCP has congestion and flow control mechanisms. Unlike SCTP, IoSL creates an ID for each sub-flow of the MPTCP connection for the fast recovery of transport layer parameters. Thus, IoSL provides support for major transport layer protocols for enhanced end-to-end connectivity.

D. ID LAYER FOR MOBILITY

In horizontal handover, when the device moves from one access point to another, IoSL freezes the transport layer connections and store the connection parameters in its ID layer. Once the horizontal handover is over, IoSL resumes the transport layer connections and restores the transport layer parameters for the quick recovery of the previous network state. In the case of vertical handover, the on-going transport layer connections are terminated, due to IP change in the network layer. When the transport layer connection gets established back after the vertical handover, IoSL recognizes the connection and restores the network state through its stored characteristics in the ID layer. Thus, IoSL's ID layer assists the transport layer for fast connection recovery in handover scenarios.

VI. MODEL FOR PERFORMANCE ANALYSIS

In this section, we describe how IoSL improves the congestion & flow control mechanisms during link layer fluctuations followed by its functionality during blockage and handovers. We will consider TCP CUBIC [22] as the transport protocol in this section for highlighting the working of IoSL.

For avoiding idle time during data flow and optimal bandwidth utilization, the window size (W) should be equal to the bandwidth-delay product (BDP) [23].

$$W \approx \frac{RTT \times bandwidth}{packetsize} \tag{1}$$

A. CROSS-LAYER CONGESTION AND FLOW CONTROL

We assume that the receive window (`rwnd`) does not limit the window size so that the available window (W) is determined only by the congestion window (`cwnd`). To calculate BDP, RTT and available bandwidth need to be estimated over the wireless link. In general, RTT is calculated with the reception of each ACK using timestamps [23]. The available bandwidth can be assessed using cross-layer communication with the lower layer. Algorithm 1 shows the pseudo-code of the IoSL congestion control algorithm that illustrates how the congestion window is updated during link-layer fluctuations.

Algorithm 1 Congestion Control Algorithm

```
1:  $cwnd$: congestion window
2:  $MSS$: Maximum Segment Size
3:  $rtt_{min}$: minimum RTT of the connection
4:  $rtt_{est}$: RTT calculated using the last ACK received
5:  $bw_{est}$: estimated bandwidth
6:  $\sigma$: current SINR value
7:  $\kappa, \mu$: threshold parameters
8:  Initialization
9:      $rtt_{min} \leftarrow \infty$
10:     $cwnd \leftarrow 1\ MSS$
11:
12: for each received ACK do
13:     $rtt_{est} \leftarrow$ estimate RTT
14:     Using cross layer information:
15:         $bw_{est} \leftarrow$ estimate data rate available at the link
16:         $\sigma \leftarrow$ get the SINR value
17:
18:     if $(rtt_{est} < rrt_{min})$ then
19:         $rtt_{min} \leftarrow rtt_{est}$
20:     end if
21:     if $(\sigma < \kappa)$ then
22:         $cwnd \leftarrow \alpha \cdot bw_{est} \cdot rtt_{min}$
23:     else
24:         Update according to the conventional algorithm
25:     end if
26:     if $(cwnd > bw_{est} \cdot rtt_{min} + \mu)$ then
27:         $cwnd \leftarrow bw_{est} \cdot rtt_{min}$
28:     end if
29: end for
```

As shown in the Algorithm (line 21), when the `SINR` values are low due to the signal fluctuation in the lower layer, we assume the wireless link to be the bottleneck and therefore, use the scaled estimated BDP value to update `cwnd`. Also, to avoid excessive buffering and packet loss, `cwnd` is not allowed to increase significantly above the estimated BDP value. Thus, IoSL regulates the congestion control parameters for better utilization of available bandwidth.

To improve the downlink throughput performance, IoSL also adjusts the value of the maximum receive window and default receive window size (`rwnd`) based on the cross-layer information. During the connection establishment, IoSL initializes the maximum receive window size based on the estimated BDP and also updates the receive window (`rwnd`) at regular intervals during the handover and blockage scenarios. Thus it controls the data flow from the sender and improves the flow control mechanism in the transport layer.

Next, we explain how IoSL predicts a signal blockage or handover. IoSL maintains a weighted average of the `SINR` values as shown below.

$$SINR_{avg}^{t} = \beta \cdot \sigma + (1 - \beta) \cdot SINR_{avg}^{t-1} \tag{2}$$

where, σ is the present value of $SINR$, $SINR_{avg}^{t}$ is the average calculated at time 't', $SINR_{avg}^{t-1}$ is the previous value of the average, β is the smoothing factor, which is set equal to 0.7 as we found it to give a good de-noising effect while having enough sensitivity to detect handovers. If the following condition is true, link disconnection is predicted.

$$(SINR_{avg}^{t} - SINR_{avg}^{t-\tau} < \gamma) \quad and \quad SINR_{avg}^{t} < \epsilon \tag{3}$$

where γ and ϵ are threshold parameters and τ is adjusted according to the channel behaviour. These parameters determine steepness of drop in $SINR$ value used for predicting a disconnection.

When IoSL is operating on the receiver side and a link disconnection is predicted, IoSL enters into a freeze mode and advertises zero receive window ($rwnd = 0$) to the sender. The sender upon receiving a zero-window advertisement, freezes its congestion window, re-transmission timers, and enters into persist mode [24]. In case of an incorrect prediction (false alarm) or after the link recovers, the receiver sends three duplicate ACKs corresponding to the last received packet to restart the connection as suggested in [25].

When IoSL is operating on the sender side, it regulates the congestion window based on the lower-layer network conditions. It stores the transport layer parameters in its socket management layer that includes `ssthresh`, `cwnd`, `rwnd`, and many more. Also, it retrieves these parameters using the unique identifier for regulating the transport layer connections. When the link condition improves or, the handover is complete, IoSL uses these stored parameters to resume or restart the connection. Thus, IoSL effectively controls the congestion state and avoids the transport layer from going into slow start during blockage or a handover event.

B. PERFORMANCE ANALYSIS

To analyse the behaviour of TCP CUBIC with and without IoSL, we consider a simple scenario, where there is a sudden link disconnection and, the link recovers to the previous state. In such cases, we can assume that the standard TCP without IoSL would enter into a fresh slow start ($cwnd = 1$). On the other hand, when IoSL is enabled, it suspends all the transport layer connections and freezes the congestion window values. As soon as the link recovers, IoSL resumes all the transport connections, restoring the congestion window values as they were before the link disruption. The response of the standard TCP and IoSL can be seen in the Fig. 7.

The additional number of data segments that IoSL sends by the time the TCP recovers to the original state can be approximately calculated. Let W be the stable congestion window size at the peak, before the disconnection. Considering that timeout occurs, TCP will start growing its window exponentially during the slow start until it hits a threshold ($ssthresh$), which is set as $0.7W$ ($beta\_cubic = 0.7$) after the timeout. After hitting the $ssthresh$ value, TCP shifts to the congestion avoidance phase where the window grows in a concave cubic fashion [22] until it reaches the peak value ($= W$).

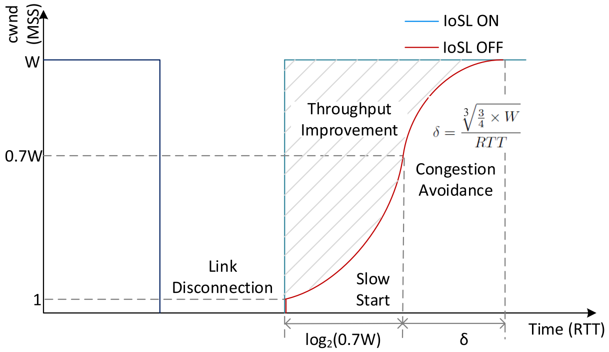


FIGURE 7. **The diagram shows the difference in response of IoSL and TCP during signal outrage. IoSL recovers instantaneously, whereas the time taken by TCP to recover can be seen to be Wlog(0.7W) + δ*RTT*. The shaded area shows the excess data sent using IoSL TCP over standard TCP.**

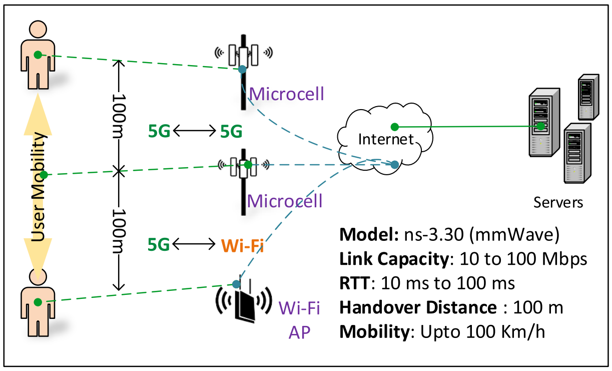


FIGURE 8. **Experimental ns-3 setup for simulating inter and intra-mobility scenario including frequent blockage for both single and double-ended modes.**

In the slow start phase, TCP will start with the congestion window of 1 MSS and double the window with each RTT. Therefore, it will take $log_2(0.7W)$ RTT for the window size to reach $0.7W$. Let the number of RTT taken in the congestion avoidance phase to reach the prior window value of W be δ. Then,

$$\delta = \frac{\sqrt[3]{\frac{3}{4}W}}{RTT} \tag{4}$$

where, we take $beta\_cubic$ = 0.7 and C = 0.4 as mentioned in [22].

We can calculate the additional data segments sent using IoSL to be:

$$W \cdot log_2(0.7W) + \frac{0.2044}{RTT} W^{\frac{4}{3}} - 1.55W - 0.08 \cdot RTT \cdot W^{\frac{2}{3}} + 1 \tag{5}$$

If we consider a frequent blockage scenario, where the next link disconnection occurs just after the standard TCP recovers from the previous disconnection, we can calculate (approximate) average throughput improvement (TI) by IoSL. For

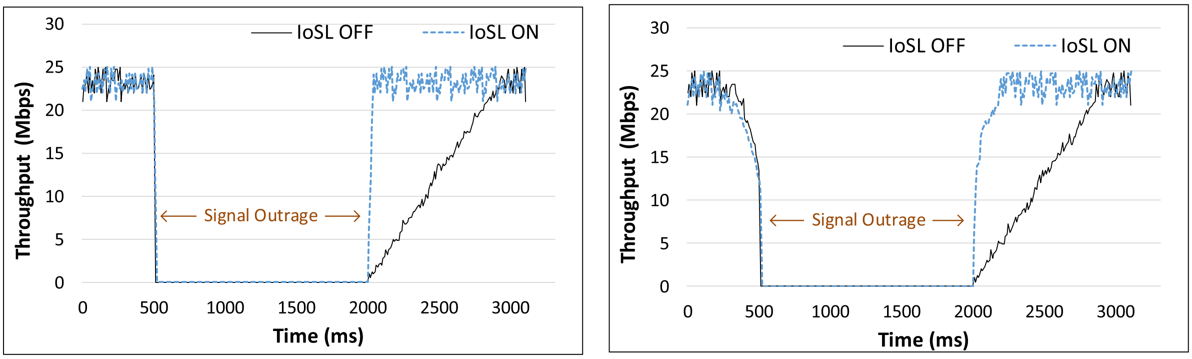


FIGURE 9. **Shows improvement in throughput and recovery time using IoSL in single ended mode during link disruptions. (a)Shows the upload scenario with IoSL on the sender side. IoSL freezes the congestion window for all transport connections and avoids slow start. (b)Shows the download scenario with IoSL on the receiver side. IoSL advertise zero window to prevent the sender from going into slow start by predicting the link disconnection sufficiently in advance.**

this, first we define the additional data as:

$$\Delta_{DT} = (W \cdot log_2(0.7W) + \frac{0.2044}{RTT} W^{\frac{4}{3}} -1.55W - 0.08 \cdot RTT \cdot W^{\frac{2}{3}} + 1) \times MSS \qquad (6)$$

where, Δ_{DT} is the amount of additional data transferred. Hence, the throughput improvement (TI) is defined as:

$$TI = \frac{\Delta_{DT}}{(log_2(0.7W) + \frac{\sqrt[3]{\frac{3}{4}W}}{RTT}) \times RTT} \qquad (7)$$

Since W is directly proportional to BDP for the end-to-end path, the above equation shows that IoSL will be more beneficial for high BDP connections. Also, as the frequency of blockage or handover increases, higher will be the improvement in throughput.

VII. PERFORMANCE EVALUATION

In this section, we provide a comprehensive analysis of the parameters discussed in Section VI to depict the significance of IoSL. We prototyped IoSL in the ns-3 simulator to evaluate the performance with multiple handovers and blockage scenarios during user mobility in Wi-Fi and 5G networks.

A. SIMULATION SETUP

The simulation setup, as shown in Fig. 8, consists of both Wi-Fi and 5G connectivity to depict various handover scenarios. To simulate inter and intra-mobility (horizontal/vertical handover) scenarios, 5G Microcell and Wi-Fi APs are configured at every 100 meters, and the user is allowed to move back and forth between any two APs, as shown in the figure. The end-user speed is allowed to vary to control the frequency and duration of the handovers. To evaluate IoSL, we have configured two different simulation scenarios and measured the performance of IoSL. These simulation scenarios consist of frequent radio signal blockages, horizontal and vertical handovers with various transport layer protocols.

B. PERFORMANCE EVALUATION ON SIMULATED ENVIRONMENT

In the first experiment, we evaluated the performance of IoSL during frequent blockages and link disconnections caused by external objects and horizontal handovers. For the experiments, two devices are configured for the evaluation - one with IoSL in single-ended mode and the other without IoSL. Further, the wireless links get disconnected due to blockages as the users move, which causes a signal outage in both the devices and results in a significant throughput drop.

As shown in Fig. 9(a), IoSL recovers the network state by regulating cwnd in the upload scenario, whereas, the device configured without IoSL takes a long time to reach network capacity due to the limitation of the congestion control mechanism. In the case of the download scenario, IoSL identifies the link disconnection and stops the data flow by sending a zero receive window $(rwnd = 0)$ advertisement to the sender. As shown in Fig. 9(b), IoSL quickly recovers the connection as the link is re-established by sending triple duplicate ACKs to the sender, whereas, the device without IoSL suffers due to packet loss and retransmission timeout. This experiment proves that IoSL efficiently regulates the transport layer parameters for better utilization of available bandwidth during blockage and horizontal handover scenarios.

In the next experiment, IoSL is evaluated in double-ended mode with vertical handover scenarios. First, the performace of IoSL is tested by varying the channel characteristics, including the end-to-end delay and channel capacity. Fig. 10(a) shows the effect of end-to-end delay on the average transport state recovery time (which is the time taken for the connection to reach 25% of the available throughput post disconnection). It is seen that the IoSL improves the average recovery time upto 39.35%. Fig. 10(b) shows the performance of IoSL as the channel capacity is varied. It can be seen that IoSL reaches the peak throughput faster as compared to standard TCP. Also, as increasing the channel capacity increases the BDP of connection, we can see that the throughput improvement with IoSL increases as inferred from Equation 7.

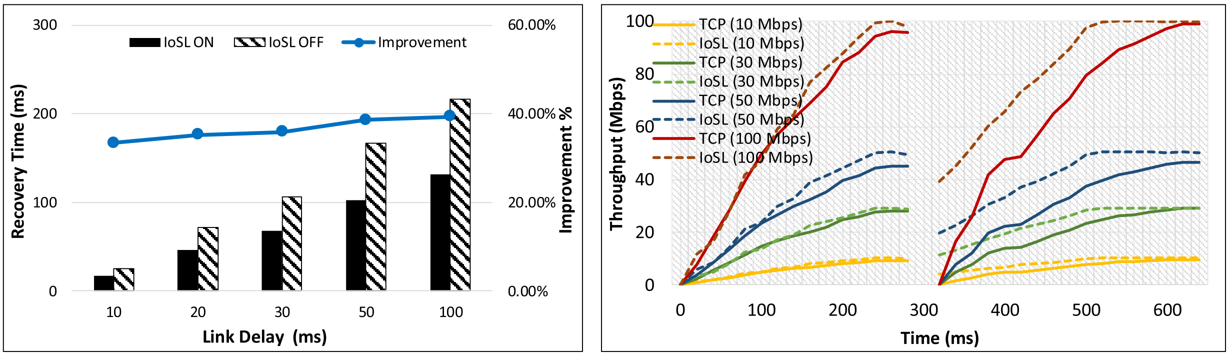


FIGURE 10. Performance Analysis of double-ended mode IoSL with varying simulation parameters. (a) shows the average transport state recovery time with RTT variation. As the link delay increases, IoSL achieves better performance in terms of reconnection and recovery time upto 39%. (b) shows the throughput graph during reconnection as the channel capacity is varied from 10 to 100 Mbps. IoSL achieves the peak throughput faster as compared to Standard TCP.

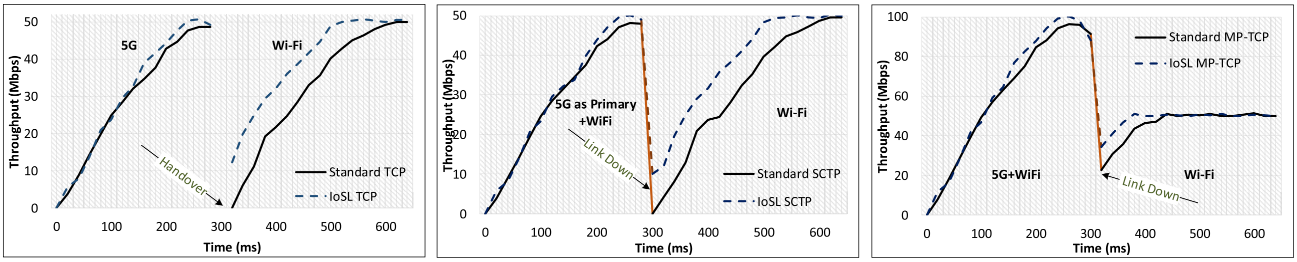


FIGURE 11. Performance Analysis of double-ended mode IoSL during vertical handover over different transport protocols - (a)TCP, (b) SCTP, and (c)MPTCP. The result clearly shows that each transport layer with IoSL, recovers the transport state much faster and restarts the new connection after handover with old network state of the connection.

Since IoSL is designed to operate on major transport layer protocols, we evaluated IoSL performance on transport protocols such as TCP, SCTP and MPTCP. Fig. 11(a) shows the fast connection recovery of TCP with IoSL in comparison with the standard TCP. In the case of standard TCP, the connection is disconnected and recovered slowly, whereas, with IoSL, the TCP connection is recreated and recovered quickly, providing better throughput. Next, to evaluate the performance IoSL over SCTP, the devices are configured to connect with 5G as primary and Wi-Fi as a secondary path initially, and then the 5G link goes down. As shown in the Fig. 11(b), the device with IoSL SCTP adjusts its transport layer parameters and recovers its network state in Wi-Fi quickly, leading to better throughput. Finally, in the case of MPTCP, the device is configured to connect with both Wi-Fi and 5G network initially, and then the 5G link goes down. As shown in the Fig. 11. (c), the IoSL device instantiates the network state much faster than the standard MPTCP. Hence, IoSL successfully adjusts the transport layer parameters and recovers much faster during the handovers.

VIII. EXPERIMENTAL RESULTS

We conducted several rounds of experiments to test IoSL performance in terms of enhanced user experience. In this section, we compared the throughput and reconnection time during mobility and handovers scenarios, with and without IoSL. The experiments are conducted in both simulated and live-air environment.

A. OVERALL PERFORMANCE IN SIMULATED ENVIRONMENT

This section includes the overall performance of IoSL in the simulated environment. In this experiment, we configured end-user device speed increases gradually with multiple handovers and analyzed the throughput performance. As shown in Fig. 12(a), when the end user's speed increases, causing more frequent handovers, IoSL provides a 12.79% improvement in throughput as compared to standard TCP. Also, we conducted several experiments to measure the performance of connection recovery time. Fig. 12(b) compares the CDF of reconnection time with and without the IoSL solution in vertical handovers (inter-network mobility). From the results, we observed that 80% of the IoSL connections take less than 30 milliseconds for reconnection, whereas the standard TCP reconnects only 50% of the connections by that time.

B. LIVE-AIR RESULTS

The proposed solution is successfully implemented in various Samsung devices, such as Note 10 and its variants with Android 10. We used two identical devices simultaneously for testing with and without IoSL and evaluated the solution by comparing the throughput improvement during vertical handovers. Both the devices were configured with the same hardware, software and network conditions. The results are taken from live-air network by switching between LTE and Wi-Fi after every 10 seconds.

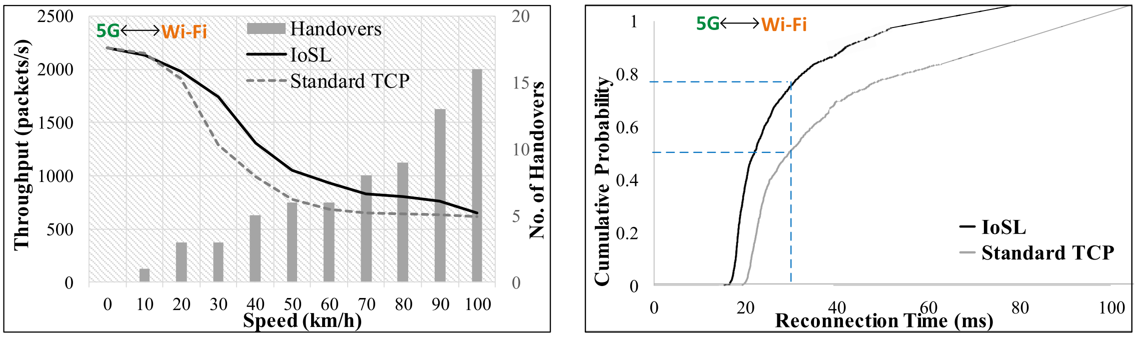


FIGURE 12. Inter-network mobility simulation using 5G and Wi-Fi. Fig. (a) shows the throughput improvement by up to 12% as the number of handovers increases. Fig. (b) compares the cumulative probability of reconnection time. IoSL reduces reconnection time up to 33% for vertical handovers.

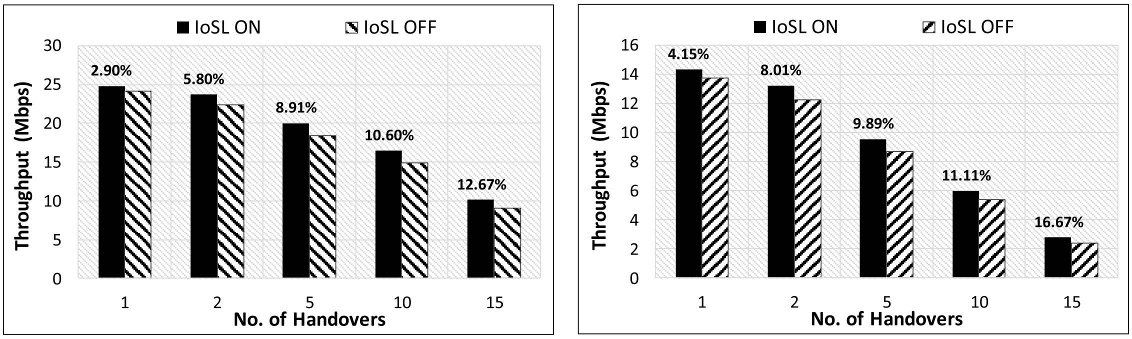


FIGURE 13. Impact of IoSL during frequent vertical handovers for file (a) download and (b) upload scenarios. The results show significant throughput improvement by up to 12.6% and 16.6% for download and upload cases, respectively.

The live-air evaluation results are based on download and upload of a 2GB file, where the maximum download speed for the operator is observed as 20 Mbps, and the upload speed varies from 5 to 12 Mbps. Fig. 13(a) shows the end-user throughput improvement achieved with IoSL in the download scenario. The results prove that IoSL improves the throughput by up to 12.67% with the number of handovers increase. Fig. 13(b) shows the impact of IoSL in the uploading case. It improves the throughput further up to 16.67% as it has better control over the data rate through the congestion window of the transport layer connections. Hence, the results prove that the end-user Quality of Experience (QoE) is improved significantly during user mobility with the IoSL solution.

IX. CONCLUSION

The futuristic network applications set mobility as one of the key requirements for achieving seamless end-to-end connectivity. However, the current transport layer is not well designed to handle network dynamics, user mobility and network handovers. Hence, in this article, we propose IoSL, an ID-based solution to guarantee seamless end-to-end communication. IoSL is a cross-layer solution that monitors the real-time network conditions as well as maintains unique identifiers for the transport layer connections to provide uninterrupted end-user experience. It is an easily deployable software solution that significantly improves performance during mobility. It is prototyped and evaluated using the ns-3 simulator as well real-time Android device, and the results show an improvement in the end-user throughput up to 16% and average reconnection latency around 33% during various mobility scenarios.

REFERENCES

[1] J. M. Ppallan, S. Jaiswal, K. Arunachalam, P. Imputato, S. Avallone, S. S. Dronamraju, and M. R. Kanagarathinam, "Reducing user perceived latency in smart phones exploiting IP network diversity," *IEEE Access*, vol. 8, pp. 143055–143065, 2020.

[2] Z. Ning, P. Dong, X. Wang, X. Hu, J. Liu, L. Guo, B. Hu, R. Kwok, and V. C. M. Leung, "Partial computation offloading and adaptive task scheduling for 5G-enabled vehicular networks," *IEEE Trans. Mobile Comput.*, early access, Sep. 18, 2020, doi: 10.1109/TMC.2020.3025116.

[3] Z. Ning, K. Zhang, X. Wang, L. Guo, X. Hu, J. Huang, B. Hu, and R. Y. K. Kwok, "Intelligent edge computing in Internet of vehicles: A joint computation offloading and caching solution," *IEEE Trans. Intell. Transp. Syst.*, early access, Jun. 5, 2020, doi: 10.1109/TITS.2020.2997832.

[4] W. Quan, K. Wang, Y. Liu, N. Cheng, H. Zhang, and X. (. Shen, "Software-defined collaborative offloading for heterogeneous vehicular networks," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–9, Jan. 2018.

[5] K. Arunachalam, J. M. Ppallan, S. Jaiswal, R. S. Lingappa, V. Balasubramanian, and K. Subramaniam, "Layer 4 accelerator (L4A) for optimizing network protocol latencies in mobile devices," in *Proc. IEEE 20th Int. Conf. High Perform. Comput. Communications; IEEE 16th Int. Conf. Smart City; IEEE 4th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Jun. 2018, pp. 439–448.

[6] J. M. Ppallan, S. Jaiswal, K. Arunachalam, D. S. Sabareesh, M. R. Kanagarathinam, P. Imputato, and S. Avallone, "NexGen connectivity optimizer: An enhancement of smart phone performance for better connectivity," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–7.

[7] D. S. Sabareesh, G. V. P. Reddy, S. Jaiswal, J. M. Ppallan, K. Arunachalam, and Y. Wu, "Redundant TCP connector (RTC) for improving the performance of mobile devices," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–7.

[8] K. Arunachalam, Y. Hong, W. Lee, and J. M. Ppallan, "Low power TCP for enhanced battery life in mobile devices," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7.

[9] K. Govindan, K. Arunachalam, and K. Subramaniam, "Optimal server selection policy for improved network efficiency in smart phones," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2018, pp. 1–6.

[10] M. Zhang, M. Mezzavilla, R. Ford, S. Rangan, S. Panwar, E. Mellios, D. Kong, A. Nix, and M. Zorzi, "Transport layer performance in 5G mmWave cellular," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2016, pp. 730–735.

[11] S. Floyd, D. K. K. Ramakrishnan, and D. L. Black, *The Addition of Explicit Congestion Notification (ECN) to IP*, document RFC 3168, Sep. 2001. [Online]. Available: https://rfc-editor.org/rfc/rfc3168.txt

[12] B. Y. L. Kimura, H. C. Guardia, and E. D. S. Moreira, "A session-based mobile socket layer for disruption tolerance on the Internet," *IEEE Trans. Mobile Comput.*, vol. 13, no. 8, pp. 1668–1680, Aug. 2014.

[13] M. Oulmahdi, C. Chassot, and N. Van Wambeke, "Transport protocols: Limitations, evolution obstacles and solutions for an actual deployment in the Internet," *Int. J. Parallel, Emergent Distrib. Syst.*, vol. 30, pp. 1–21, Nov. 2015.

[14] J. M. Ppallan, K. Arunachalam, S. Jaiswal, D. S. Sabareesh, S. Seo, and M. R. Kanagarathinam, "Flare-DNS resolver (FDR) for optimizing DNS lookup overhead in mobile devices," in *Proc. 16th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2019, pp. 1–7.

[15] A. Sutton, R. Li, P. P. Esnault, G. Foster, L. Han, E. G. Gras, M. Shepherd, J. Sheng, K. Makhijani, J. Grant, M. Mohajer, D. R. Lopez, J. Day, C. Yue, and Yinyue, "Next generation protocols—Market drivers and key scenarios," Eur. Telecommun. Standards Inst. (ETSI), Sophia Antipolis, France, White Paper 17, May 2016. [Online]. Available: https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp17_Next_Generation_Protocols_v01.pdf

[16] B. Da, R. Li, and X. Xu, "ID oriented networking (ION) for IoT interoperation," in *Proc. Global Internet Things Summit (GIoTS)*, Jun. 2017, pp. 1–5.

[17] A. Venkataramani, J. F. Kurose, D. Raychaudhuri, K. Nagaraja, M. Mao, and S. Banerjee, "MobilityFirst: A mobility-centric and trustworthy Internet architecture," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 74–80, Jul. 2014.

[18] K. Yu, S. Eum, T. Kurita, Q. Hua, T. Sato, H. Nakazato, T. Asami, and V. P. Kafle, "Information-centric networking: Research and standardization status," *IEEE Access*, vol. 7, pp. 126164–126176, 2019.

[19] T. Goff, J. Moronski, D. S. Phatak, and V. Gupta, "Freeze-TCP: A true end-to-end TCP enhancement mechanism for mobile environments," in *Proc. IEEE INFOCOM. Conf. Comput. Commun., 19th Annu. Joint Conf. IEEE Comput. Commun. Societies*, Mar. 2000, pp. 1537–1545.

[20] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar, and J. Bailey, "The quic transport protocol: Design and Internet-scale deployment," in *Proc. Conf. ACM Special Interest Group Data Commun.*, Aug. 2017, pp. 183–196.

[21] T. Azzino, M. Drago, M. Polese, A. Zanella, and M. Zorzi, "X-TCP: A cross layer approach for TCP uplink flows in mmwave networks," in *Proc. 16th Annu. Medit. Ad Hoc Netw. Workshop (Med-Hoc-Net)*, Jun. 2017, pp. 1–6.

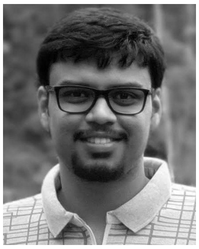
[22] I. Rhee, L. Xu, S. Ha, A. Zimmermann, L. Eggert, and R. Scheffenegger, *CUBIC for Fast Long-Distance Networks*, document RFC 8312, Feb. 2018. [Online]. Available: https://rfc-editor.org/rfc/rfc8312.txt

[23] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 6th ed. London, U.K.: Pearson, 2012.

[24] M. Bashyam, A. Ramaiah, and M. Jethanandani, *TCP Sender Clarification for Persist Condition*, document RFC 6429, Dec. 2011. [Online]. Available: https://rfc-editor.org/rfc/rfc6429.txt

[25] R. Caceres and L. Iftode, "Improving the performance of reliable transport protocols in mobile computing environments," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 5, pp. 850–857, Jun. 1995.

KARTHIKEYAN ARUNACHALAM (Senior Member, IEEE) received the B.Tech. degree in information technology from Anna University, Chennai, India. He has more than 15 years of extensive research experience in transport layer protocols. He was a Senior Associate with Novell Software Development (India) Pvt. Ltd., a Senior Software Engineer with Huawei Technologies India Pvt. Ltd., and a Software Engineer with Protechsoft Technologies Pvt. Ltd. He is currently an Architect with the Samsung R&D Institute India-Bangalore. His current research interests include next-generation transport layer protocols, cross-layer communication, and mobile edge computing. He is a member of ACM.



SHIVA SOUHITH GANTHA (Member, IEEE) received the B.Tech. degree in electrical engineering with minors in computer science and engineering from the IIT Bombay, India, in 2019. He is currently an Engineer with Samsung R&D Institute India-Bangalore. His research interests lie in area of communication and networking, including next-generation networks, transport, and network protocols.

SWETA JAISWAL (Member, IEEE) received the B.Tech. degree in electronics and communication from the Vellore Institute of Technology, Vellore, India. She has more than 11 years of experience in software research and development in the telecommunication industry. She was a Software Engineer with Tata Consultancy Service Ltd. She is currently a Chief Engineer with the Samsung R&D Institute India-Bangalore. Her research interests include communication and networking, including next-generation transport layer protocols, multi-access edge computing, green communication techniques, and cross-layer optimization.

JAMSHEED MANJA PPALLAN (Member, IEEE) received the B.Tech. degree in computer science from the Cochin University of Science and Technology, India, in 2012. He has more than eight years of industry experience in software research and development. He is currently working as a Research Engineer with the Samsung R&D Institute India-Bangalore. He was a Senior Software Engineer for Huawei Technologies India Pvt. Ltd., and an Associate Programmer for the National Informatics Centre, Government of India. His research interests include next-generation transport and network layer protocols, cross-layer optimization, smart phone operating systems, and green communication.

SEONGKYU SONG (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from The University of Texas at Austin, Austin, in 2005. He is currently a Principal Engineer with Samsung Electronics, where he is participating in research and development of next generation telecommunication systems. His research interests include network architecture and protocols, programmable networking, and next generation telecommunication networks.

ANSHUMAN NIGAM (Member, IEEE) received the bachelor's degree in electrical engineering from IIT Kanpur. While at Samsung India, he has actively worked in research, development, and standards for 5G, LTE and WiMAX technologies. He has served as an active contributor from Samsung to 3GPP RAN2 and WiMAX2.0 standards groups from 2008 to 2014, and led the 5G Research and PoC Development Team, SRIB, with a primary focus on creation of new MAC and higher layers technologies along with the development of high-speed data path for 5G on state-of-the-art chipsets. He is currently a Principal Architect with Samsung R&D Institute India-Bangalore (SRIB), where he is leading the 6G Research Team. His current research interest lies in conceptualization of the technologies for beyond 5G and 6G systems, particularly in higher terahertz bands and in realization of the practical applications of AI in wireless systems.