A Method for Enabling Context-Awareness at Transport Layer for Improved Quality-of-Service Control

JAMSHEED MANJA PPALLAN¹, (Member, IEEE), KARTHIKEYAN ARUNACHALAM³, (Senior Member, IEEE), SHIVA SOUHITH GANTHA¹, (Member, IEEE), SWETA JAISWAL³, (Member, IEEE), SEONGKYU SONG², (Member, IEEE), AND ANSHUMAN NIGAM¹, (Member, IEEE)

¹6G Laboratory, Samsung Research and Development Institute India-Bangalore, Bengaluru 560037, India
²6G Research Team, Samsung Seoul Research and Development Campus, Seoul 06765, South Korea

Corresponding author: Jamsheed Manja Ppallan (jamsheed.mp@samsung.com)

ABSTRACT

As the 5G systems have introduced network slicing on the virtualized network environment, application-specific QoS (Quality of Service) management has come to the fore to support a range of services, such as eMBB (enhanced Mobile Broadband), URLLC (Ultra-Reliable Low Latency Communications), and mIoT (massive Internet of Things). It is expected for the next-generation systems to support more diverse applications demanding newer and granular quality characteristics for network services. Most of the current transport layer protocols, built on layering design and best-effort paradigm, do not provide the necessary precision and adaptability required for various QoS support. Therefore, we need new protocols that comprehend application behaviours and adapt to dynamic network conditions for more fine-grained QoS enhancement. This paper presents a Context-oriented Transport (CoT) layer for the next-generation network that understands the application context and adapts to varying network conditions with flow-based QoS control. CoT is an end-to-end software solution that improves the underlying network capacity utilization and prioritizes the traffic flows according to quality needs. We prototyped CoT in Linux/Android devices and evaluated the performance with emulated traffic environment. The experiments show that CoT reduces latency by up to 16.5% and improves average throughput by up to 30%, compared with Android 10 in LTE network, resulting from enhanced traffic classification and network utilization.

INDEX TERMS

Cross-layer method, inter-layer optimization, machine learning, quality of service, QoS control, TCP/IP, traffic classification, traffic control, transport protocols.

I. INTRODUCTION

The research focus towards the 6G system has begun by defining the futuristic Key Performance Indicators (KPI’s), while the evolutionary 5G deployment is well underway. The next-generation networks are set to go beyond mobile Internet to support ubiquitous services and applications ranging from Virtual Reality (VR) to Extended Reality (XR), Holoportation, Teleportation, and Digital Twins [1]. Further, the Beyond 5G (B5G) era is expected to provide a data rate as high as 1 Terabit per second (Tbps) and ultra-low latency up to microseconds along with ubiquitous connectivity.

Hence, it requires a system that supports flexible and dynamic provisioning of resources with guaranteed QoS to support these stringent application requirements along with massive connectivity and capacity.

For instance, as Samsung published 6G white paper [2], High-Precision Network (HPN) is suggested as an enabling technology to achieve these extreme QoS requirements. Although the upcoming technologies are still evolving, the futuristic applications mandate the presence of HPN with deterministic end-to-end latency, which also set acceptable jitter to nanosecond-level. Moreover, the emerging services demand bandwidth guarantee along with ultra-low latency. Hence, Network slicing and traffic shaping for guaranteed end-to-end QoS [3] are evolving to address these
diverging performance requirements that many verticals impose in terms of latency, scalability, availability, and reliability.

The increasing demand for QoS requirements due to the emerging services has led to various network-layer mechanisms’ evolution for cellular (5G QoS Identifier) [4] and internet (Traffic Class/Type of Service) [5] models. 5QI is a mechanism standardized by the 3GPP cellular network to classify packets into different QoS classes. Also, the network layer mechanisms such as IntServ and DiffServ provide QoS services at the network layer. However, the QoS mapping between the network layer and the application layer is still a challenge. The transport layer, which sits between the application and network layers, is one of the best targets to enable fine control and improve the end-to-end QoS. Moreover, it must be redesigned to incorporate the external parameters and network environment aspects to achieve these crucial requirements. At present, the transport layer manages every flow uniformly and maintains best-effort fairness for all the applications, as shown in Fig. 1. However, every application has its own QoS expectation, such as constant bit-rate, high-data-rate, delay-tolerant, or high reliability. Thus, the current transport layer can create a bottleneck in optimally catering to the QoS needs of all the applications.

![FIGURE 1. QoS control bottleneck of transport layer.](image1)

Hence, there is a need for a self-evolving protocol suite that responds dynamically to the context of application traffic and the network conditions. Considering all these limitations of the existing transport layer, we propose an end-to-end software solution named Context-oriented Transport (CoT), which could be the first step towards the next-generation transport layer. CoT provides an ML-based classification solution, which comprehends the application context, evaluates network conditions, prioritizes traffic flows, and configures protocol parameters to achieve fine-grained QoS control.

II. MOTIVATION

The exponential growth of smart devices in the coming decades will result in a huge increase in mobile data traffic. According to the survey [6], the number of connected devices may rise to 50 billion, and data traffic may reach five zettabytes (ZB) by 2030 [7], as shown in Fig. 2. Along with these increasing numbers, the emergent applications and services have much more stringent requirements for latency and throughput, which will exceed the expectations of 5G. This section discusses the technological advancements and limitations that motivated the perception of our proposal.

A. NEED FOR APPLICATION-AWARE QoS CONTROL

The current transport layer fails to consider applications’ quality requirements and limits the possibility of application-aware QoS enhancement. However, network slicing is emerging as a promising technology for the 5G network to support flexible and on-demand provisioning of physical resources using a virtualized network environment [8]. It can fulfill the diverse network requirements by supporting the customized configuration of resources, management models, and system parameters for various application use cases. Hence, along with network slicing, solutions to enable the application-specific QoS and traffic shaping are also crucial for the future network to facilitate a wide variety of verticals.

B. DRAWBACKS OF END-TO-END QoS MECHANISMS

Even though many cellular and other network technologies are available for QoS control, some barriers restrict the benefits from these technologies from reaching the applications on the end host. For instance, 3GPP has defined various 5QI values as stated in [9], which explains the QoS profile for any service. Since cellular networks rely on the notion of bearers, the QoS is provided at the bearer level. In practice, a Guaranteed Bit Rate (GBR) bearer is set up for VoIP flows. And since it is difficult to identify precise QoS requirements for the internet data flows, they are sent using a default Non-Guaranteed Bit Rate (Non-GBR) bearer. Along with that, many network layer mechanisms, such as Integrated Services (IntServ) and Differentiated Services (DiffServ) [10] help to achieve low latency for critical network traffic while maintaining best-effort service for non-critical services. However, these QoS control features without cross-layer information are inadequate for futuristic applications with stringent quality needs. Also, they are not explored by the application developers for better QoS. Diff-Serv mechanism helps the bottlenecked router decide which packets to drop first. It will indirectly control the sending rate
at the transport layer at the end hosts. As we will see later, CoT responds faster by directly controlling the sending rate according to the QoS classes when the bottleneck shifts to the wireless access link.

C. EVOLUTION OF DETERMINISTIC NETWORKING

DetNet working group of IETF [11] is working on defining Deterministic Networks, which will enable unicast/multicast data flows with bounded latency and jitter within a single network domain. DetNet operates at the network layer and delivers service over lower-layer technologies such as Multi-Protocol Label Switching (MPLS) [12], and IEEE 802.1 Time-Sensitive Networking (TSN) [13]. Even though DetNet is being deployed in real-time operational applications, it is isolated from external access due to many unsolved challenges in the current 5G network [11]. Enabling deterministic flows over wireless links is a challenging task due to possible large variations in channel capacity. As the channel quality degrades, all the non-deterministic flows will be first to suffer. Priority-based classification of even the non-deterministic flows would help provide a better Quality of Experience to the end-user over varying link capacity. Hence, flow-based traffic shaping and QoS control using inter-layer optimization is crucial for achieving end-to-end guaranteed service delivery.

D. THE TRANSPORT LAYER ‘FAIRNESS’ BOTTLENECK

The current transport layer, designed based on best-effort delivery, manages every application uniformly and maintains fairness, which may hamper the applications’ expectations and user experience. It is responsible for achieving reliability and performing congestion and flow control for an end-to-end transmission. However, the transport layer requirements for future applications, in terms of throughput, data rate, and latency get more stringent with time, which will be difficult to achieve with existing protocols. Hence, it is crucial to design a protocol suite that provides flexibility to dynamically adapt to the application requirements and the network conditions to achieve the KPI’s of future networks.

To deal with these limitations of the transport layer, we designed and implemented Context-oriented Transport (CoT) that performs fine-grained QoS enhancements for the application flows. The experimental results prove that flow-based QoS control using protocol parameter modification and network layer prioritization improve the user experience significantly.

E. NOVELTY AND IMPORTANCE

- Context-oriented Transport is a novel approach with a flow-based traffic shaping mechanism, which dynamically modifies transport layer parameters based on precise QoS requirement.
- CoT is an end-to-end software solution that is compatible with the majority of existing transport layer protocols and does not require any change in the middleboxes or network infrastructures.
- It uses effective methodologies for ML-based traffic classification using connection characteristics, transport layer customization and flow prioritization to enable QoS control at the transport layer.
- Our experiments show that CoT improves QoS performance significantly, which results in reduced latency by up to 16.5% and improved throughput by up to 30%.

III. BACKGROUND AND RELATED WORK

With the emergence of futuristic applications and technologies, achieving end-to-end QoS has gained lots of attention. Many research works revolve around QoS for 5G New Radio (NR). 3GPP has also introduced standard 5QI values [4], as mentioned in section II, to identify a specific QoS forwarding behaviour for a 5G QoS Flow based on Priority Level, Packet Delay Budget and Averaging Window. Namwon et al. also proposed a slice management scheme in [14] that mitigates the wireless interference among slices by the prioritized interference-aware routing and admission control. In the SDN side, S. Khairi et al. have proposed a solution [15], which integrates IP Multimedia Subsystem and the SDN to improve the Quality of Service (QoS) management in the network side. However, there are few constraints, which hampers the overall performance in terms of capacity utilization. Similarly, an SDN based multipath solution is proposed in [16], to achieve QoS through class-specific bandwidth guarantee. It divides the traffic into different classes and forwards them into three different queues with pre-configured rates.

Along with these developments, many research works are going around QoS management at the transport layer. Zhani in FlexNGIA [17] has analyzed the characteristics and requirements of future networking applications. It has pointed out a few major limitations of the TCP/IP suite and proposed a solution to fix them, but still, there are many unsolved challenges. Similarly, S. Shi et al. presents Deadline-aware Transport Protocol (DTP) [18] to provide deliver-before deadline service. Here, the application needs to communicate the metadata and deadline of the data to DTP. In addition to these solutions, there are new congestion control algorithms such as Data Center TCP [19] that uses Explicit Congestion Notification (ECN) [20] to provide window-based control methods. Similarly, IATCP [21] is a rate-based congestion control approach that counts the total number of packets injected to meet the Bandwidth-Delay Product (BDP) of the network dynamically. QUIC [22] is a user-space transport protocol that solves some of the drawbacks of TCP, such as connection establishment overhead, head of line blocking and IP mobility. However, similar to TCP, QUIC also treats all traffic equally and does not introduce any QoS mechanisms at the transport layer. These solutions are confined to specific use cases and services, and also, need changes in applications and network infrastructures.

Since CoT performs traffic classification, we look into the recent work involving traffic classification. In the literature, the traffic classification problem has been mainly looked at from the network perspective. Due to increased privacy
concerns and encryption, traffic classification has become a challenging task. Many of the recently proposed mechanisms show that Machine Learning and Deep Learning methods achieve good results. The work presented in [23] surveyed Deep Learning-based classification models for network traffic classification. [24] uses machine learning techniques to identify the services within HTTPS connections. Giuseppe Aceto et al. perform traffic classification on a smartphone for identifying the mobile apps [25]. The feature set used in these works includes flow characteristics which can vary depending on the access network. For example, some statistics like mean and variance of inter-packet arrival time will be affected by the access network to which the end-host is connected. But CoT includes additional features using cross-layer information for better classification accuracy.

Hence, in this proposal, we considered all these limitations and proposed CoT for attaining guaranteed QoS by performing per-flow traffic shaping. CoT is completely different from these earlier works and supports majority of the transport layer protocols.

IV. DESIGN AND ARCHITECTURE

A. OVERVIEW

Context-oriented Transport (CoT) is an end-to-end software solution which adopts the inter-layer optimization technique to provide fine-grained QoS control at the transport layer. CoT, which follows an ML-based classification algorithm, understands application features and evaluates the network state to prioritize the traffic flows based on the QoS expectations. CoT is a transparent solution, which is compatible with the existing transport layer protocols.

B. SOFTWARE ARCHITECTURE

CoT is a shim layer between the application and transport layer which consists of three major modules, namely, Context Analyzer, Traffic Classifier and QoS Manager, as shown in Fig. 3. Context Analyzer analyzes the flow characteristics using application and transport layer attributes and, network characteristics using several lower layer attributes. Then, Traffic Classifier performs traffic classification and forms QoS classes with various quality requirements according to the application behavior. As shown in Fig. 4, it uses the application/flow characteristics provided by Context Analyzer as input (\(X\)) and generates the QoS classes as output (\(y = h(X)\)) (refer to Section V-B). Hence, Traffic Classifier makes the transport layer capable of understanding the flow context.

FIGURE 3. Software architecture of Context-oriented Transport. CoT is a shim layer between the application and transport layer which adopts the inter-layer optimization technique to understand the QoS requirements of the application layer and underlying network.

FIGURE 4. The diagram depicts the functional aspects of Context Analyzer, Traffic Classifier and QoS Manager. The operations include the feature extraction of incoming flow, classification based on QoS-need, analysis of network condition, and implementation of QoS control.

C. MODULAR DESIGN AND OVERALL OPERATIONS

This section explains the functional perspectives of the three modules of CoT and its overall operation flow, as shown in Fig. 4.

1) CONTEXT ANALYZER: PROVIDING INTER-LAYER AWARENESS

Context Analyzer follows a cross-layer mechanism to monitor the application-level and transport layer flow-level characteristics using the attributes listed in Section V-A. Also, it collects the real-time connection interface and network attributes from lower layers for assessing the network condition. Thus, Context Analyzer aims to bridge the gap between the layers above and below the transport layer.

2) TRAFFIC CLASSIFIER: COMPOSING APPLICATION CONTEXT

Traffic Classifier performs traffic classification and forms QoS classes with various quality requirements according to the application behavior. As shown in Fig. 4, it uses the application/flow characteristics provided by Context Analyzer as input (\(X\)) and generates the QoS classes as output (\(y = h(X)\)) (refer to Section V-B). Hence, Traffic Classifier makes the transport layer capable of understanding the flow context.
3) **QoS MANAGER: ACHIEVING FINE-GRAINED QUALITY-OF-SERVICE**

QoS Manager utilizes the QoS classes created by Traffic Classifier ($y$) and evaluates the network condition using Context Analyzer’s lower-layer attributes ($\vec{W}$). Based on the network capability evaluation, it decides the flow control parameters for enhanced quality control and flow prioritization ($z_k = f(y, \vec{W})$). Thus, QoS Manager overrides the drawbacks of best-effort paradigm and casts the first stone for the next-generation transport layer.

4) **OPERATION FLOW**

The operation flow of CoT begins with the arrival of a data packet from client-application, extraction of its features, classification based on QoS-need, analysis of network condition, and implementation of QoS control and flow prioritization. Firstly, as depicted in Fig. 4, Context Analyzer recognizes and extracts the transport layer connection information. Later, it performs the feature selection to create the data set for classification. Traffic Classifier uses the classification model, identifies the flows that require QoS control and fits them into various QoS Classes for distinguishing the quality demands of the flows. Based on the QoS classes, QoS Manager evaluates the network condition and applies QoS control for creating the prioritized traffic according to the QoS classes. Also, it validates the ongoing flows and provides feedback for reinforcing the classification learning model.

V. **LEARNING MODEL AND ALGORITHMS**

CoT utilizes a supervised machine learning classification algorithm for creating flow-based QoS classes. The trained classifier is used to categorize the live flows into appropriate QoS classes. The QoS Manager acts on these classes based on the network parameters. This section will explain the functional details of the Traffic Classifier and the QoS Manager.

A. **DATA SET AND FEATURE SELECTION**

We extract the application-related information, network traces and radio layer information to generate the training data. Most of the prior ML-based network traffic forecasting methods use only flow-based statistics as features. Since these statistics change for various access networks, the classification accuracy of such models will be affected when the UE is operating over a different network or even when the network conditions largely vary. Therefore, we include additional network-state features captured from the lower layers. The training dataset is created by combining network traces collected using Samsung devices over various access networks (LTE, Wi-Fi) over four weeks at multiple locations. Fig. 5 depicts the heterogeneity of the data set used for training in terms of their duration and data-transferred. We do pre-process to ensure that the data set will not overfit our model. The training data is then split according to the type of access network. We train a different classifier for each network type as the set of features available differs for various access networks. Then the flows, which need quality control, are filtered out and, the data set ($D$) is generated by labelling them into various QoS blocks. Table 1 lists the connection and network features used for training our classification model.

**TABLE 1. Flow and network-specific features for the proposed traffic classification model.**

| Feature | Flow-specific features | Description |
|---------|------------------------|-------------|
| 5-Tuple | Source/destination IP address, port, and transport protocol |
| Connection Duration | Duration of a transport layer connection |
| Data Sent/Received | Amount of data sent/received per flow |
| Connection Frequency | Pattern of connection establishment and termination |
| Read/Write Periodicity | Pattern of application’s read and write from the buffers |
| Flow Name | A domain name associated with the flow such as URL, FQDN, CNAME, SNI |
| Service Name | Mapping of dedicated port numbers to specific service |
| Unique App Identifier | UID for distinguishing an app |
| RAT | Radio Access Technologies - LTE, 5G/Wi-Fi |
| Throughput | Bandwidth available to the end device |
| RTT | Round-Trip Time of the network |
| Packet Loss Ratio | Measurement of congestion in the path. |
| RSSI, ASU Power, CQI | Signal quality measurement |

B. **PROPOSED TRAFFIC CLASSIFICATION MODEL**

This section focuses on the machine learning-based algorithm used for traffic classification. We use Random Forest [26] classifiers for classifying the flows into their QoS classes. We use Random Forest as it is shown to perform well in traffic classification tasks [24] and is faster due to its less real-time execution complexity.

1) **RANDOM FOREST MODEL**

CoT implements a Random Forest (RF) model, an ensemble of multiple Decision Tree estimators. The number of Decision Tree estimators for the RF model is set as 10 for each access network type. The Decision Trees are trained by a random selection of features and bagging techniques. We train
an additional Decision Tree based on features available at the flow initialization time. Any new flow that needs to be classified is given as input to each of the decision trees, which give their outputs. The RF model chooses the QoS profile which is chosen by a majority of trees and assigns that class to the flow. During live operation, it might happen that certain features are not available for the flow. In such cases, we only consider the outputs of the Decision Trees, which do not use that feature for classification. If there is no such tree, then the flow is categorized into the best-effort class. Let us consider a given data set $D$ with size $n$,

$$D = \{ (\tilde{X}_1, y_1), (\tilde{X}_2, y_2), \ldots, (\tilde{X}_n, y_n) \}$$

(1)

where, $\tilde{X}$ be the feature vector and $y$ be the predicted outcome, which is drawn from a probability distribution $(\tilde{X}_i, y_i) \sim (\tilde{X}, Y)$. Each feature vector contains the set of features used for defining the Decision Tree

$$\tilde{X}_k = \{ x_{k1}, x_{k2}, \ldots, x_{kp} \}$$

(2)

The classifier $h$, which predicts $y$ from $(\tilde{X})$ based on the data set of examples $D$, can be represented as an ensemble of classifiers.

$$h = \{ h_1(\tilde{X}), h_2(\tilde{X}), \ldots, h_K(\tilde{X}) \}$$

(3)

We define the parameters of each decision tree $h_k(\tilde{X})$ to be

$$\theta_k = \{ \theta_{k1}, \theta_{k2}, \ldots, \theta_{kp} \}$$

(4)

Hence, $h_k(\tilde{X})$ can also be represented as $h_k(\tilde{X}|\theta_k)$

Fig. 6 shows an example of a partition of a two-dimensional feature space by recursive binary splitting of data using duration ($x_1$) and amount of data transferred ($x_2$) attributes, to create RF of Decision Trees. Based on $x_1$ and $x_2$, short-lived ($h_1(\tilde{X}|\theta_1)$), long-lived ($h_2(\tilde{X}|\theta_2)$), light-weight ($h_3(\tilde{X}|\theta_4)$), and best-effort subsets are created. Further, connection creation ($x_3$) and domain name matching ($x_4$) split the subsets into QoS classes such as Streaming ($h_4(\tilde{X}|\theta_4)$), light-weight ($h_3(\tilde{X}|\theta_3)$), heavy-weight ($h_4(\tilde{X}|\theta_4)$) and best-effort subsets are created. For classification, the given classifiers. Moreover, it also reflects the confidence in the classification. The larger the margin, the more the confidence.

2) QoS CLASSES
The example in the previous paragraph considers only four classes for simplicity of the explanation. CoT classifies the flows into 25 classes based on four different characteristics of the flows. As these characteristics are independent, we train a different RF classifier for improving accuracy and reducing the complexity of the Traffic Classifier module. CoT categorizes a flow into the “best-effort” class if it cannot be classified otherwise due to unavailability of some feature data or when the confidence margin ($\hat{m}(\tilde{X}, y)$) is too low. The QoS manager does not tune these flows. Next, we mention the flow characteristics based on which the QoS classes are defined.

3) ELASTICITY
determines if a flow can adjust to the varying throughput and delays in the end-to-end path and still be able to meet the needs of the application. Each of the flows is categorized between Flexible-Bit-Rate (FBR) and Constant-Bit-Rate (CBR) classes. Flows that are categorized as CBR cannot adapt to varying network conditions without affecting the application’s quality requirements. Some examples of CBR traffic are interactive gaming, VoIP flows. Whereas a good example of FBR traffic is multimedia streaming. As the multimedia flows are buffered before being played, they can tolerate variations in the throughput and delay. In case of a persistent throughput variation, the server reacts by changing the resolution of the multimedia content and adjusts the transmission rate. Also, non-interactive type traffic or bulk transfers can be considered as FBR with High-Data-Rate requirements.

4) DELAY TOLERANCE
specifies how much the given flow is tolerant to increase in the end-to-end delay. Based on the level of sensitivity
to increased delays, the flows are classified into three categories – Low-Latency, High-Reliability and Non-Critical. Low-Latency flows such as interactive gaming need strong bounds on the delay and, hence, an increase in delay affects the quality. High-Reliability flows correspond to the which are impacted by an increase in delay but are not as sensitive as Low-Latency flows. Instant messaging is an example of High-Reliability flow. The flows which are not affected by the increase in the delay come under Non-Critical flows. All background application flows like regular backup, updates, and a few more come under this class.

5) DATA RATE
helps to differentiate between data heavy flows and light flows. These flows are classified into Thin and Thick classes respectively.

6) FLOW DURATION
The flows are categorized based on their duration into Short and Long flows.

It is important to note that it is possible for a flow to change its behaviour with time. Therefore, the Traffic Classifier periodically updates the classes for all the flows.

C. QoS CONTROL ALGORITHM
QoS Manager is responsible for performing quality control after the traffic classification. Algorithm 1 describes the QoS control logic for network quality assessment, which utilizes the network parameters ($\hat{W}$).

$$\hat{W} = \{w_1, w_2, \ldots, w_m\}$$ (7)

where, $m$ be the total number of network parameters used.

QoS Manager broadly divides the network parameters into two; signal strength parameters and transmission quality parameters. Similar to [27], signal quality of the underlying network can be estimated using RSSI, RSSNR, and CQI attributes. Also, various transmission quality parameters such as latency, throughput and packet loss ratio are considered for evaluating the transmission quality. QoS Manager estimates the network condition and decides the QoS enhancement concerning the current network state.

Based on the network capability and the QoS class of the flow, QoS Manager decides the flow control parameters such as initial congestion window, slow start threshold (ssthresh), receive-window (rwnd), and congestion window (cwnd). Also, it sets the IP header flags such as Type of Service (IPv6) and DSCP (IPv4) to take advantage of the QoS mechanisms deployed in the network nodes.

QoS Manager keeps track of per-flow and per class transmission quality parameters – latency, throughput and packet loss ratio that get updated periodically. Also, the QoS Manager maintains other relevant statistics for each class. For example, an exponential moving average of the cwnd of all flows belonging to that class is maintained. The main goal of the QoS manager is to provide the best end-user experience given the QoS classes of the flows and the current network condition. There are three major actions of the QoS manager which are explained below.

1) SETTING THE INITIAL CONGESTION WINDOW
The TCP flows start with a small initial congestion window size (up to 10 MSS [28]) when in the slow start phase. The initial window is a critical parameter, especially for short duration flows that end before transitioning to the congestion avoidance state. Therefore, the flow completion times for short-flows are greatly dependant on the initial window size. Since the QoS manager keeps per class statistics of the average cwnd, it can choose a better initial window. It is important to note that as this action needs to be taken at the start of the flow, the classifier is relying on partial information like the 5-tuple and service name for assigning a QoS class.

2) BANDWIDTH REDISTRIBUTION
When the network condition becomes poor, the QoS manager checks if the overall throughput of the flows has reduced as compared to the previous value. If yes, it assumes that the bottleneck for the majority of flows has shifted to the wireless access link. QoS manager kicks in and tries to redistribute the bandwidth by reducing the data rate of FBR, Non-Critical flows. QoS manager achieves this for the uplink flows by limiting the cwnd. For the downlink flows, it modifies the rwnd field sent in the ACKs to limit the transmission rate at the sender side. As a result, the High-Reliability and CBR flows

---

**Algorithm 1** Network State for QoS Control

1: $\hat{W}$: Network Parameters
2: netstate: [excellent, good, avg, poor]
3: iface: connection interface
4: $\sigma$: Signal Quality
5: $\phi$: Transmission Quality
6: $\tau, \epsilon$: threshold parameters
7: $\alpha, \beta$: coefficients of variation
8: Initialization
9: iface $\leftarrow$ [LTE/5G/Wi-Fi]
10: $\hat{W} \leftarrow$ iface, $\sigma, \phi$
11:
12: while iface $\neq$ down do
13: if ($\sigma < \tau$) then
14: netstat$_{\sigma} \leftarrow$ [avg, poor]
15: else
16: netstat$_{\sigma} \leftarrow$ [excellent, good]
17: end if
18: if ($\phi < \epsilon$) then
19: netstat$_{\phi} \leftarrow$ [avg, poor]
20: else
21: netstat$_{\phi} \leftarrow$ [excellent, good]
22: end if
23: netstate $\leftarrow$ $\alpha \cdot$ netstat$_{\phi} + \beta \cdot$ netstat$_{\sigma}$
24: return netstate
25: end while
will capture the excess bandwidth available. Hence, by providing more bandwidth for CBR/HDR and Low-Latency flows, CoT tries to provide a better user experience as the network conditions vary.

3) RETRANSMISSION TIMER ADJUSTMENT
The retransmission timer decides the amount of time to wait until a packet is considered lost and is retransmitted. A timer that is too short will cause unnecessary retransmission and will waste the bandwidth resources. When the network quality becomes bad, and the available data rate is meagre, unnecessary retransmissions will further reduce the available bandwidth. In such scenarios, the QoS Manager slightly increases the retransmission timer value for FBR, Non-Critical flows to avoid unnecessary retransmissions clogging the bandwidth resources.

These mechanisms not only prioritize the critical flows but also improve the overall utilization. Since CoT understands and monitors the changing network conditions, it can react faster to congestion leading to improved resource utilization.

VI. PERFORMANCE EVALUATION
This section discusses the design challenges and performance evaluation of CoT. Firstly, we discuss the accuracy of the classification algorithm achieved during the training and testing periods. Then, the emulated traffic environment created for evaluating CoT performance is described with the proposed topology. Also, we evaluate the performance metrics of the QoS classes.

A. DESIGN CHALLENGES
The performance evaluation methods assess the transport layer customization and flow prioritization techniques of CoT. The proposed traffic classification and QoS control algorithms are designed to improve the end-user experience. We have considered the following challenges and limitations of the design and network infrastructure for the performance evaluation.

1) QoS NEEDS FROM FLOW CHARACTERISTICS
It is challenging to extract the QoS requirements of an application from a set of application and transport layer parameters. However, using a very vast and versatile data set, the ML model of CoT is trained well for making optimal decisions to classify the flows into multiple QoS buckets. CoT odds out the connection that do not require QoS control, using the duration/data/frequency pattern along with the domain name and service information. Segregation of critical and non-critical flows makes it easier to perform the traffic classifications.

2) DYNAMIC NETWORK CONDITIONS
The network state algorithm proposed in CoT does not consider the dynamically fluctuating wireless network behaviour. It, on the other way, focuses on the current state of the network using pre-defined network state labels such as poor, avg., good, and excellent. CoT makes use of the network state for customizing the transport layer parameters.

3) NETWORK PATH FEATURES
Even though the cellular network introduces various flow prioritization techniques to comprehend QoS control, the public internet domain still depends on DiffServ and ToS mechanisms. DSCP requires its functionality support all over the path. Hence, we consider the entire route to support DiffServ and ToS for the performance evaluation.

4) END-HOST COMMUNICATION
The CoT needs to modify both the end-hosts for enhancing congestion control and flow control (sender and receiver). Also, the network state algorithm, which runs at the wireless link, is not considered at the remote-host for transport layer customization.

B. TRAINING AND TESTING STRATEGY
We evaluate the performance of the classification model using the data set mentioned in Section V-A. Fig. 7 shows our prediction methodology, where the data set gets divided into training and testing parts. During the evaluation phase, outputs are predicted using the inputs (the test data) and, the error between predicted and actual values is calculated.

The mean absolute percentage error (MAPE) is one of the most widely used measures of forecast accuracy due to its scale-independent and easily interpretable nature.

\[
E_i = \begin{cases} 
1, & \text{if } y_i \neq j_i \\
0, & \text{otherwise} 
\end{cases} 
\]

\[
MAPE = \left( \frac{1}{N} \sum_{i=0}^{N} E_i \right) \times 100
\]
where, $E_i$ be the misclassification index, which will be set to 1 if the classification is wrong, $MAPE$ be the Mean Absolute Percentage Error and $N$ be the number of flows classified. Fig. 8 depicts the $MAPE$ of the model with randomly distributed QoS classes. CoT classification has a $MAPE$ of 4.48% to 9.4% when the training data set size varies from 1000 to 10,000. It signifies that the training data used is not over-fitting to the RF model.

**FIGURE 8.** Mean Absolute Percentage Error (MAPE) of the classification model. When there is a misclassification of a flow into wrong QoS class, we consider the error as $E = 1$. Our model has a MAPE of 4.5% to 9.4% when using a data set of up to 10,000 entries with all the QoS classes.

### C. EMULATED TRAFFIC SETUP

We have prototyped CoT in Linux environment with Ubuntu 18.04 and deployed in end-devices for performance evaluation. The emulation setup is configured using Network Emulator (NetEm) [29] for performing traffic shaping, as shown in Fig. 9. NetEm is a tool used for regulating packet losses and delay into network traffic with traffic-control utility. As the network conditions vary with delay, packet loss and available throughput, the performance of CoT also varies in terms of throughput improvement and latency reduction. Multiple network scenarios are configured by varying link delay from 0 to 50 ms and packet loss ratio from 0 to 2% and, conducted experiments to evaluate QoS improvements.

**FIGURE 9.** Emulated traffic setup for performance evaluation. End-devices with Linux (Ubuntu 18.04) are configured with/without CoT solution and NetEm is used for emulating traffic for evaluating various network conditions.

### D. EVALUATION OF QoS CLASSES

The impact of QoS control on the QoS classes is estimated using the experimental setup with multiple flows of different quality requirements at the same time. To ensure that the CoT enhancements are improving the performance of the given flows, we portrayed data-rate, latency and network utilization metrics in respect of their QoS profiles.

Fig. 10 shows the bandwidth redistribution between a Non-Critical and a CBR flows. The experiment is conducted with varying the signal quality of the network, which affects the throughput. As marked in the figure, the achieved goodput of Non-Critical flows reduced drastically with the network condition. But, CoT ensures that the CBR flows get sufficient throughput, with bandwidth redistribution, even when the available throughput drops due to bad network conditions.

**FIGURE 10.** Effect of CoT on different QoS classes as the network condition turns bad. CoT redistributes bandwidth and ensures that critical, CBR traffic gets required throughput.

Next, Fig. 11 illustrates the latency reduction for Low-Latency flows during the network packet loss conditions. CoT minimizes the impact of network conditions and improves the goodput by 40%.

**FIGURE 11.** Latency reduction for High-Reliability flows during the network packet loss conditions. CoT minimizes the impact of network conditions and improves the goodput by 40%.

VOLUME 9, 2021
network conditions. Though the packet loss differs up to 2%, CoT minimizes the impact of network conditions and improves the goodput by 40%.

Finally, Fig. 12, shows the transport layer parameters modification for High-Data-Rate (HDR) flows. The RTT of the link is varied from 5ms to 50ms for showing the impacts of latency on HDR flows. CoT modifies the buffer parameters and configures high initial receive windows such as $90 \times$ MSS, $120 \times$ MSS, and $180 \times$ MSS. The simulation proves that increasing the initial window improves the achieved throughput even when the network conditions degrade (with increased latency).

![Figure 12](image12.png)

**FIGURE 12.** CoT modifies the buffer parameters and configures with high initial receive windows to improve the data rate for HDR flows.

### VII. EXPERIMENTAL RESULTS

This section discusses the experimental results captured using the emulated traffic setup for network delay and packet loss simulation and Android devices using Live-air LTE network. CoT utilizes the network capacity effectively and improves the user-experience significantly, in terms of latency reduction and throughput improvement.

#### A. EMULATED TRAFFIC EVALUATION

The experimental setup was configured with a maximum of 100 Mbps bandwidth capacity and 20 ms as the default RTT. Also, packet loss ratio up to 2% and link delay up to 50 ms are infused for producing various network situations. We define link delay or simulated delay as the additional latency induced by the simulator in the end-to-end path for varying the RTT. This section discusses throughput improvement, latency reduction and increased network capacity utilization for several QoS classes.

Fig. 13 analyses the characteristics of Low-Latency and CBR flows during the lossy network state with link delay. The experiment infuses packet losses and link delay simultaneously for varying the network conditions. CoT improves the throughput by up to 30% for keeping the higher data rate. Fig. 14 shows the actual latency comparison of High-Reliability and Latency-sensitive flows with delay simulation. CoT reduces user-perceived latency by up to 16.5% for improving the reliability even during worst network conditions.

![Figure 13](image13.png)

**FIGURE 13.** Impact of CoT on end-user experience improvement. Analysis of HDR and CBR flows during the lossy network state with link delay. CoT reduces the drop in data-rate.

![Figure 14](image14.png)

**FIGURE 14.** The actual latency comparison of High-Reliability and Latency-sensitive flows with delay simulation. CoT reduces user-perceived latency by up to 16.5% for improving the reliability even during worst network conditions.

![Figure 15](image15.png)

**FIGURE 15.** CoT improves bandwidth utilization on an average to 30% for satisfying the QoS requirements even during high end-to-end delay and congestion.

improving the reliability even during worst network conditions.

Then, network capacity utilization is evaluated with an ideal network scenario created using a single UE associated
with gNodeb at a time, having 100 Mbps capacity. Hence, we estimate the bandwidth utilization with the percentage of network capacity attained by the UE as the goodput (when goodput is 50 Mbps, 50% is the utilization). In an ideal scenario, throughput degradation due to packet losses is considered as the change in network load. As the packet loss ratio varies, the achievable network capacity degrades similarly to variation of network load. CoT ensures optimal bandwidth utilization on an average of 28%, by notifying the network layer for prioritization and modifying parameters to meet the QoS requirements even during high end-to-end delay and congestion, as shown in Fig. 15.

B. LIVE-AIR EVALUATION

The proposed solution is successfully implemented in various Samsung smart phones, such as Note 10 variants and A-series models with Android 10. We used two identical devices simultaneously for testing with and without CoT and evaluated the solution by comparing different network quality parameters. Both the devices were configured with the same hardware, software and network conditions. The results are taken from a live-air LTE network with general usage pattern taken from a live-air LTE network with general usage pattern.

We considered various application categories with different QoS profiles and evaluated by running the apps simultaneously for ensuring diversity in results. The experiments show that CoT precisely classifies application flows into appropriate categories. The test traffic includes E-commerce (Best-Effort), Video Call (CBR), Online Gaming (High-Reliability and Low-Latency) and Streaming (HDR, FBR). flows along with background downloads. As shown in Fig. 16 CoT improves the throughput of these applications by up to 30.9% and reduces the average latency by up to 23.15% when compared with the default Android platform using LTE live-air network in India.

VIII. CONCLUSION

The futuristic Internet applications are set to unleash its wide variety of services with the upcoming 5G/6G networks. However, the current transport layer fails to acknowledge the precise QoS requirements of these applications, which affects the performance and hampers the user experience. CoT provides flow-based QoS control to the transport layer that responds dynamically and flexibly to the applications' characteristics and network conditions. CoT consists of an ML-based algorithm for classifying applications and their connections into different QoS classes based on various features selected. We prototyped and evaluated CoT using Linux/Android environment with emulated network conditions. It consistently improves throughput and reduces latency by up to 30% and 16.5% respectively. Also, the results using Android devices with live-air LTE network proves that CoT precisely categorizes application flows according to their QoS profiles.

REFERENCES

[1] A. Mourad, R. Yang, P. H. Lehne, and A. de la Oliva, “Towards 6G: Evolution of key performance indicators and technology trends,” in Proc. 2nd 6G Wireless Summit (6G SUMMIT), Mar. 2020, pp. 1–5.
[2] Samsung. (Jul. 2020). Samsung’s 6G Vision White Paper. Samsung Research. [Online]. Available: https://research.samsung.com/next-generation-communications
[3] B. Pinczel, D. Gebberger, Z. Turanyi, and B. Formanek, “Towards high performance packet processing for 5G,” in Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN), Nov. 2015, pp. 67–73.
[4] 5G; NR; Overall Description; Stage-2, Standard (TS) 38.300, 3GPP, Version 15.5.0, 3rd Generation Partnership Project (3GPP), Tech. Specification, Apr. 2019.
[5] O. J. S. Parra, A. P. Rios, and G. L. Rubio, “IPV6 and IPV4 QoS mechanisms,” in Proc. 13th Int. Conf. Inf. Integr. Web-Based Appl. Services (iWAS), 2011, pp. 463–466.
[6] ITU-R. (May 2016). Mobile Traffic Forecast. Internation Telecommunication Union. [Online]. Available: https://www.itu.int/dms_pub/itu-r/rep-R-REP-M.2370-2015-PDF-E.pdf
[7] F. Tariq, M. R. A. Khan, K.-K. Wong, M. A. Imran, M. Bennis, and M. Debbah, “A speculative study on 6G,” IEEE Wireless Commun., vol. 27, no. 4, pp. 118–125, Aug. 2020.
[8] M. Jiang, M. Condoluci, and T. Mahmoodi, “Network slicing management prioritization in 5G mobile systems,” in Proc. 22th Eur. Wireless Conf. Eur. Wireless, May 2016, pp. 1–6.
[9] System Architecture for the 5G System (5GS), Standard 23.501, Version 16.5.0, 3GPP, 3rd Generation Partnership Project (3GPP), Tech. Specification (TS), Jul. 2020.
[10] I. Mahadevan and K. M. Sivalingam, “Quality of service architectures for wireless networks: IntServ and Diffserv models,” in Proc. 4th Int. Symp. Parallel Archit., Algorithms, Netw. (ISPAA'99), Jun. 1999, pp. 420–425.
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 is currently working as an Architect with Samsung Research and Development Institute India-Bangalore. Previously, he worked as 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. 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 minor in computer science and engineering from Indian Institute of Technology Bombay, Maharashtra, India, in 2019. He is currently working as an Engineer with Samsung Research and Development Institute India-Bangalore. His research interests include 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 Vellore Institute of Technology, Vellore, Tamil Nadu, India. She has more than 11 years of experience in software research and development in the telecommunication industry. She is currently working as a Chief Engineer with Samsung Research and Development Institute India-Bangalore. Previously, she worked as a Software Engineer with Tata Consultancy Services Ltd. Her research interests include communication and networking which includes next-generation transport layer protocols, multi-access edge computing (MEC), green communication techniques, and cross-layer optimization.

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

ANSHUMAN NIGAM (Member, IEEE) received the degree in electrical engineering from IIT Kanpur. He is a Principal Architect, associated with Samsung Research and Development Institute India-Bangalore (SRIB), where he is currently leading the 6G Research Team. While at Samsung India, he has actively worked in research, development, and standards for 5G, and LTE and WiMAX technologies. From 2008 to 2014, he has served as an active contributor from Samsung to 3GPP RAN2 and WiMAX 2.0 Standards Groups. Then he 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 the state-of-the-art chipsets. His current research interests include 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.