

# Path Aware Transport Layer Solution for Mobile Networks

SHIVA SOUHITH GANTHA<sup>1</sup>, (Member, IEEE), SWETA JAISWAL<sup>1</sup>, (Member, IEEE),  
JAMSHEED MANJA PPALLAN<sup>1</sup>, (Member, IEEE),  
AND KARTHIKEYAN ARUNACHALAM<sup>1</sup>, (Senior Member, IEEE)

Next Generation Networks, Samsung Research and Development Institute, Bengaluru 560037, India

Corresponding author: Shiva Souhith Gantha (shiva.gantha@samsung.com)

**ABSTRACT** The revolutionary Fifth Generation (5G) network technology has already been deployed in many countries to provide end-to-end quality-of-service for a wide variety of services, applications, and users with extremely diverse requirements. To accomplish the same, 5G architecture has enabled Multi-Connectivity (MC) in mobile devices, which aggregates various radio resources and enables simultaneous connectivity. Moreover, the current mobile devices are equipped with multiple network interfaces (Wi-Fi/Cellular) as well as dual-stack networks (IPv4/IPv6) support. But, their performance degrades significantly due to variation in network path quality. The current design of the Internet protocol suite has not adapted well to utilize these multiple interfaces for providing better network reliability. Hence, we propose a client only software solution called Path Aware Transport (PAT), which explores communication paths between client and server and adapts to the dynamics of the network to migrate from one Transmission Control Protocol (TCP) connection to another without hampering the user experience. PAT is prototyped and evaluated in 5G using the ns3 simulator. Moreover, it is implemented in the Samsung flagship mobile device with Android Pie. PAT significantly improves the content downloading time up to 20% and the page loading time up to 29% consistently.

**INDEX TERMS** Content distribution networks (CDN), cross-layer design, modeling and optimization, wireless and mobile networks performance.

## I. INTRODUCTION

5G and Beyond 5G (B5G) networks aim to empower various next-generation network applications such as immersive eXtended Reality (XR), remote surgery, autonomous vehicles, and smart industry. For this, future networks should provide high end-user throughput and ultra-low latency with high reliability. To achieve these high data rates, 5G utilizes the mmWave spectrum, and B5G will move towards the terahertz spectrum [1]. However, since mmWave signals have low range and are highly susceptible to blockage, channel quality on mmWave links can be extremely fluctuating [2]–[4]. These problems may be more severe in the higher frequency terahertz spectrum.

Present mobile devices have multiple network interfaces (Wi-Fi/Cellular) and also support Multi-Connectivity (MC) to improve service quality with parallel connection to multiple cells, as shown in Fig 1. However, the network path

quality may vary due to varying signal strength, congestion, and type of network, which results in multiple handovers. Hence, it is important to find a transport layer solution to maximize user experienced throughput during network quality variation.

TCP remains the leading transport layer protocol for reliable end-to-end communication. Mobile Network Operators have improved network infrastructures to support high data rates and reduced latencies but could not minimize the transport layer connectivity overheads. The operators cannot minimize these user-perceived delays and connectivity overheads [5], [6] without making any modifications in TCP. TCP has evolved considerably over last few decades. The current design of TCP is not adapted well to cope with the changes in lower layers, communication paths, and multi-path diversity.

Considering all these limitations, we propose a novel solution called Path Aware Transport Layer (PAT) solution for mobile devices. As the name implies, it explores the available communication paths between User Equipment (UE) and content servers. Based on the estimated path quality,

The associate editor coordinating the review of this manuscript and approving it for publication was Weisi Guo<sup>1</sup>.

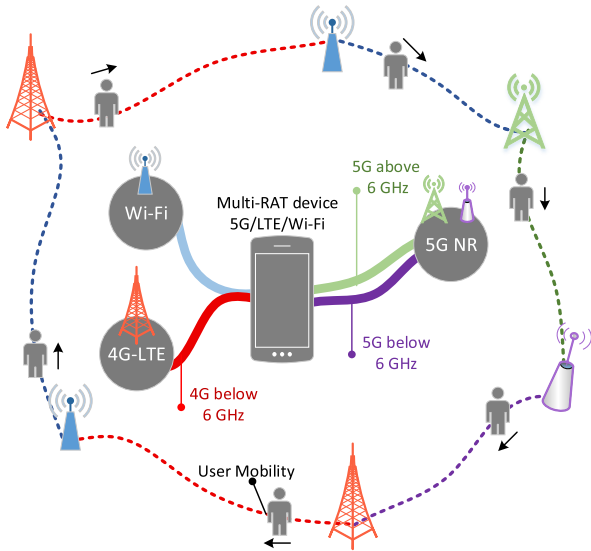


FIGURE 1. Multi connectivity of Wi-Fi/LTE/5G NR.

it dynamically migrates one TCP connection to another and also navigates from one content server to another without affecting the user experience.

The remainder of the paper is organized as follows. Section II discusses about the motivation for our proposal. In Section III, we compare our proposal with some of the related works. Section IV describes the design and architecture of PAT and its sub modules. Section V discusses about implementation challenges and solution for this work. Section VI explains the mathematical model which is used for performance analysis. Section VII gives the performance analysis for PAT. Section VIII discusses our experimental results and Section IX concludes the paper with future work.

II. MOTIVATION

The choice of end-to-end connection paths can significantly impact the service quality and user experience. Our simple experiments show that the performance depends on the selected path and server. We conducted experiments in dynamic network conditions, including Wi-Fi and LTE, to analyze the RTT of the content servers of popular multimedia streaming applications such as Netflix and Amazon Video. Our experiment consisted of two similar devices, where the first device always connect to the server with the lowest RTT (called the *best server*), and second to the *default server*, which corresponds to the default entry in the DNS response. We observed that the RTT of the content servers varies due to the latency of network interface, communication path quality, and processing time of the server. Fig 2 shows the average throughput comparison between the default and the best server for some popular video streaming applications taken over more than 100 trials. The result proves that the default server chosen by the applications is not always the best, and there is a great scope for improvement of the performance.

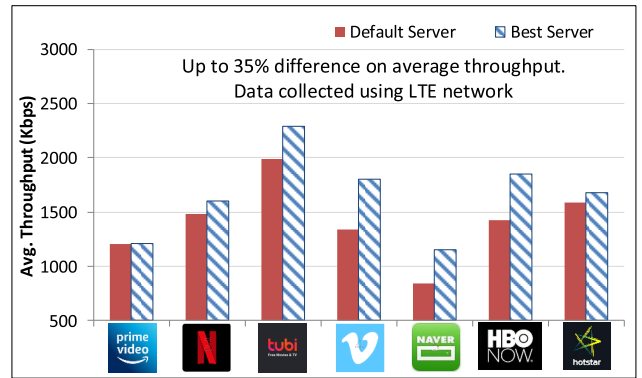


FIGURE 2. Our study on throughput comparison of popular applications shows that servers chosen by default are not the best ones available.

A. APPLICATION IS NOT AWARE OF THE BEST SERVER AT THE MOMENT

Content Distribution Network (CDN) is the prime technology that provides an infrastructure to distribute web content across multiple servers at different locations. In general, these web servers are referred with various domain names. The local or operator DNS servers do not expose all content servers to the client applications. The typical behavior of the client is to choose the first IP address from the DNS response over the subsequent IP addresses [7]. But this default server, as chosen by the client, is not always the best server at the moment for download [8].

B. DYNAMIC VARIATIONS IN QoS

User mobility, variable signal strength, handover, or congestion in the network can result in fluctuation of the available bandwidth. Current smart devices are equipped with multiple network interfaces (Wi-Fi, 3G/4G/LTE, and 5G NR) and support dual-stack networks (IPv4/IPv6). An alternate available network interface might offer better throughput to the end-user. We wish to utilize these multiple interfaces to provide the best QoS to the end-user.

C. EASILY DEPLOYABLE SOLUTION

Deployment of new transport layer solutions is the most restrictive problem in the evolution of the transport layer [9]. Solutions that require changes in the servers or the intermediate nodes would have deployment issues in scale and in time as well as backward compatibility. Client only solutions are easy to deploy and scale effectively.

Considering the above-explained problems, we designed a client-based solution, Path-Aware Transport (PAT) layer solution, which explores the network path’s quality and finds the best network path for the transport layer. The solution is implemented in Android and Linux platforms, and the performance is measured using real-time packet trace analysis. The results consistently show an improvement in the download time up to 20% and page loading time up to 29%.

#### D. NOVELTY AND CONTRIBUTION

- PAT is the state of the art and client-only software solution which explores all available CDN servers and finds the best communication path at the moment.
- PAT is a proprietary solution that discovers the CDN infrastructure with the help of multiple DNS servers and network interfaces. It migrates TCP connections seamlessly between available interfaces (Wi-Fi/LTE) based on the network path quality.
- PAT is implemented as middleware and does not require any change in applications, network protocols, kernel, middleboxes or servers.

#### III. RELATED WORK

In literature, Kim *et al.* proposed a solution named mHTTP [10], which exploits the network path diversity and establishes simultaneous TCP connections with multiple servers to fetch a single content. Another literature [11] is also similar, which utilizes path diversity and source diversity to stream high-quality videos from the server. Reference [12] is also a related idea, where a chunk request is created and fetched simultaneously. However, all these works are limited to HTTP downloads and do not switch the path dynamically.

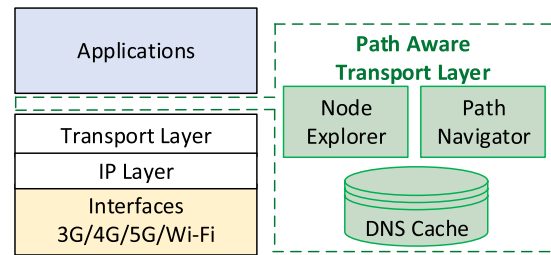
Some other works like [13] and [14] aim to reduce the DNS lookup time by sending DNS queries to multiple DNS servers in parallel and using the quickest response. Our observation shows that the fastest DNS response does not always correspond to the nearest available mirror server. In [15] a DNS-proxy based solution for CDN server selection is proposed, but it does not support switch over in the middle of the connection, when the best available path is changed in case of user mobility. In [16] and [17], another DNS-based solution is proposed, which exploits IP network diversity for improving the connectivity. But, for heterogenous networks with high mobility, exploiting only IPv4 and IPv6 paths will not be sufficient to make an impact.

Multipath TCP (MPTCP), the standard protocol proposed by the IETF in [18], is a solution that tries to leverage the path diversity available between a single client and a server. However, it requires kernel-level modifications at both the client and the server which makes it difficult to deploy. Further, in [19] and [20], the authors have analyzed the problem of CDN server selection in the MPTCP context. They have even suggested extending the DNS protocol to allow the CDN infrastructure to learn that a client has multiple network interfaces. Similarly, in [8], the authors describe a solution to select the optimal mirror server for downloading a content replicated over multiple servers, but they did not experiment on the network path diversity at the client device. All these improve the CDN server selection, but none of them consider all the factors together to find the best server at any moment dynamically.

#### IV. PAT: DESIGN AND ARCHITECTURE

PAT is the first client only solution that uses all active network interfaces (Wi-Fi/Cellular) to explore the available

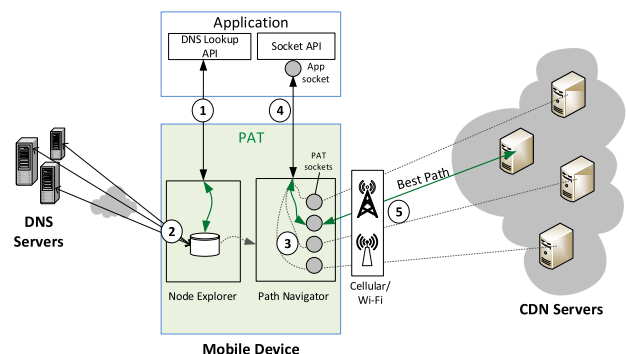
communication paths between the User Equipment (UE) and content servers. Based on the estimated path quality, it dynamically migrates between the active network interfaces without affecting the user experience.



**FIGURE 3. Architecture of PAT. It operates between the application and transport layers with two major modules - Node Explorer and Path Navigator.**

#### A. OVERVIEW

As shown in Fig 3, PAT is a client-based solution, which lies between the application and transport layers. It consists of two modules, namely, Node Explorer and Path Navigator. It also defines wrapper function for `getaddrinfo` and connect from `libc` library. The Node Explorer explores a list of content servers' IP addresses by using the local, operator, and public DNS servers. And Path Navigator finds the best network path irrespective of network type or interface and migrates these TCP connections from one mirror server to another transparently for the applications.



**FIGURE 4. Overview of PAT Operations. It discovers, monitors and migrates to the best path and mirror server dynamically.**

Fig 4 summarizes the overall operation flow of PAT. As shown in Fig 4, ① When the application tries to download content from a server, ② Node Explorer discovers all the content servers by creating a list of IP addresses from multiple DNS servers. ③ Then the Path Navigator establishes simultaneous TCP connections with the content servers through available interfaces (Wi-Fi/Cellular). ④ When the application sends data requests to the server, ⑤ the Path Navigator maps application socket descriptor to PAT's socket descriptor transparently and routes data requests through the best communication path. Moreover, during an ongoing connection, if a better path or server is detected, the Path Navigator seamlessly migrates the current TCP connection to the newly found best path and continues the download.

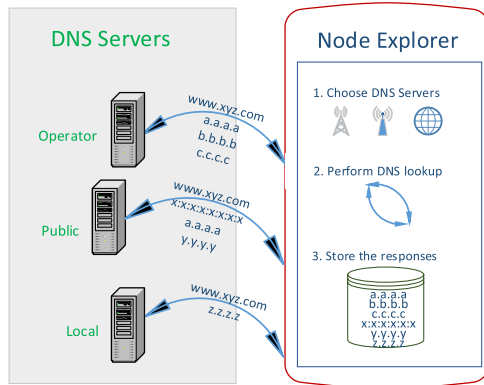


FIGURE 5. Node Explorer explores available content servers by querying available local, operator and root level DNS servers.

**B. NODE EXPLORER**

As the name implies, Node Explorer explores available content servers by querying multiple DNS servers. As shown in Fig 5, it discovers the available local, operator, and root-level DNS servers. Node Explorer sends DNS queries to available DNS servers to identify the CDN servers and stores the responses into its DNS cache. It also analyzes each DNS response and categorizes the type of network stack (IPv4/IPv6). Periodically, it renews its cache and deletes the rare or unused domain names from its cache. It also pre-resolves the domain names for the applications and performs query optimization.

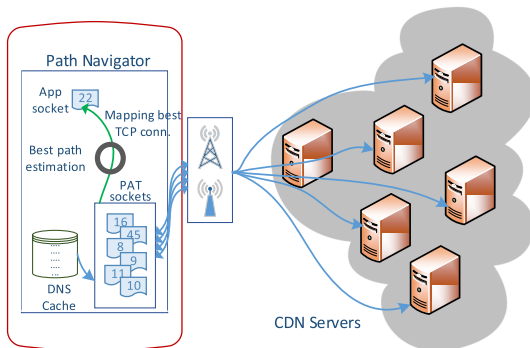


FIGURE 6. Path Navigator finds the best communication path from the explored content servers and migrates the TCP connections to the best-estimated connection transparently.

**C. PATH NAVIGATOR**

The main aim of Path Navigator is to find the best communication path from explored content servers and migrate the TCP connections to the best-estimated connection transparently without hampering the user experience. As shown in Fig 6, Path Navigator takes input from DNS Cache and identifies the list of mirror servers available for a domain name, considering both the network stack (IPv4 and IPv6). It establishes connections to these mirror servers using multiple interfaces (Wi-Fi, LTE/3G). Path Navigator monitors the RTT of TCP connections and dynamically finds the best network path, network interface, and content server available at any moment. When an application tries to establish

a new TCP connection or sends data requests, Path Navigator migrates the connection to the best-estimated TCP connection and improves the content downloading/page loading time. Algorithm 1 explains the best path exploration logic, which migrates the application connection requests to the best-estimated connection without hampering the user experience.

**Algorithm 1** Best Path Navigation Logic

```

1: ip: IP Address of the content server
2: iface: Connection Interface (enabled by the user)
3: function CreateConnection(ip, iface)
4:   while (ip != NULL) do
5:     while (iface != NULL) do
6:       Create TCP connection over iface
7:       Store the connection FD as PATFD
8:       Iterate to next available interface from iface
9:     end while
10:    Iterate to next address of ip from cache
11:  end while
12:  return PATFD
13: end function

1: PATFD: FD created by PAT
2: appFD: Application FD
3: function ExploreBestPath(PATFD, appFD)
4:  if (appFD has READ or WRITE) then
5:    while (PATFD != NULL) do
6:      Send keep-alive packet for PATFD
7:      Estimate the RTT - PATRTT
8:    end while
9:  end if
10:  if (PATRTT < appFDRTT) then
11:    Map PATFD to appFD
12:  end if
13: end function

```

**V. IMPLEMENTATION CHALLENGES**

In this section, we discuss the potential challenges associated with implementing the PAT solution in the mobile device and provide a solution for each challenge.

**A. SUPPORTING HTTPS CONNECTIONS**

In mobile devices, the application uses both HTTP and HTTPS sessions for content download. In case of secure HTTPS session, SSL/TLS certificates are exchanged between client and server in addition to TCP 3-way handshake. The application uses SSL session related APIs for secure data exchanges. PAT taps the SSL libraries and maps connected socket descriptors along with SSL parameters such as cipher suite, private/public keys, and certificates to the application [21]. So, whenever application requests for a secure connection, PAT establishes an HTTPS connection over multiple interfaces and dynamically maps the SSL session along with socket descriptor to the application.



## B. POOR NETWORK QUALITY AND CONNECTION MIGRATION

The dynamic variations in network path quality impact the performance of TCP connection and hamper the user experience. To solve this problem, PAT monitors the RTT of every TCP connection and restricts applications from establishing a TCP connection over the poor quality network interface. It also establishes redundant TCP connections over multiple interfaces and migrates the TCP connections to the best network path seamlessly, especially for long-lived streaming TCP connections.

## C. EXCESS TRAFFIC, RESOURCE WASTAGE AND POWER CONSUMPTION

PAT establishes simultaneous connections with multiple content servers, monitors the RTT of these connections, and keeps the lowest RTT connection alive. It sends probe packets to servers periodically, to keep these connections alive. The number of these probe packets is nearly insignificant as compared to the application data packets being transferred. We have verified the same with experiments, and the results are captured in Section VIII. Also, this periodic sending or receiving data from the network, may wake up the radio resources of the modem and consume device power significantly [22]–[24] and, [25]. But, PAT is carefully designed to avoid any extra radio wake up in the device. It always combines its data or socket creation requests with at least one of the application request.

## VI. MODEL FOR PERFORMANCE ANALYSIS

As discussed in the Section II, each domain name is associated with multiple content servers of varying RTTs. An application without PAT chooses the first IP address from DNS response, referred to as *default server*, for content download. Whereas, an application with PAT finds the best content server dynamically. Let us call this estimated content server as *best server*.

Let  $p_i$  be the probability that (for an application without PAT) the *default server* is same as *best server* for domain name  $i$ .

Let  $X_i$  be a random variable defined as the number of trials in which *default server* is same as *best server* for domain  $i$ .

Since, all the trials are independent of each other,  $X_i$  follows a Binomial distribution.

$$X_i \sim B(n_i, p_i) \quad (1)$$

Hence, the probability of PAT benefiting an application  $k_i$  times out of  $n_i$  trials for a given domain name  $i$  can be given by

$$Pr(X_i = n_i - k_i) = \binom{n_i}{n_i - k_i} p_i^{n_i - k_i} (1 - p_i)^{k_i} \quad (2)$$

Let  $Z_i$  be the random variable defined as the frequency ratio for a domain name  $i$

$$Z_i = \frac{X_i}{n_i} \quad (3)$$

Since  $X_i$  is a Binomial distribution, using the central limit theorem [26], we can approximate  $X_i$  (and consequently  $Z_i$ ) to a Normal distribution. Consider a set of  $D$  random domain names. Assuming their  $X_i$ s to be independent and identically distributed (IID) and taking equal number of trials (i.e.  $n = n_i \forall i$ ) we can say that  $Z_i$ s  $\forall i \in (1, D)$  are identically distributed to a random variable  $Z$  which is normally distributed.

$$Z \sim N(\mu, \sigma^2) \quad (4)$$

where,

$N$  is a normal distribution with mean  $\mu$  and variance  $\sigma^2$

We further try to quantify the improvement achieved by PAT for an application using a parameter, RTT ratio  $\gamma_{j,i}$ , defined, for  $j^{\text{th}}$  trial for a given domain  $i$ , as the ratio between the latency or RTT of the *best server* and the *default server*

$$\gamma_{j,i} = \frac{RTT_{best}^{j,i}}{RTT_{default}^{j,i}} \quad (5)$$

where,  $RTT_{best}^{j,i}$  is the RTT of *best server* in  $j^{\text{th}}$  trial for domain name  $i$ .  $RTT_{default}^{j,i}$  is the RTT of *default server* in  $j^{\text{th}}$  trial for domain name  $i$ . The average percentage gain of RTT for a domain name  $i$  can be given by

$$\delta_{avg}^i = \frac{1}{N_i} \sum_{j=1}^{N_i} \left( \frac{1}{\gamma_{j,i}} - 1 \right) \times 100\% \quad (6)$$

The average percentage gain of RTT over all the domain names for an application can be given by

$$\Delta_{RTT}^{avg} = \frac{1}{D} \sum_{i=1}^D \delta_{avg}^i \quad (7)$$

Although the latency is a crucial metric for a short-lived connections, for long-lived streaming connections throughput measurement is of more significance. Studies on throughput analysis of widely deployed TCP congestion control algorithms (like TCP NewReno [27], TCP CUBIC [28]) show that throughput is inversely proportional to RTT [29].

$$throughput \propto \frac{1}{RTT} \quad (8)$$

Thus we can approximate the corresponding throughput ratio  $\tau_i$  as

$$\tau_{j,i} = \frac{1}{\gamma_{j,i}} \quad (9)$$

$$\Delta_{tp}^{avg} = \frac{1}{D} \sum_{i=1}^D \left[ \frac{1}{N_i} \sum_{j=1}^{N_i} (\tau_{j,i} - 1) \right] \times 100\% \quad (10)$$

where  $\Delta_{tp}^{avg}$  represents the average percentage gain in throughput/goodput for an application over a set of  $D$  domain names. The metrics defined here are used to analyse the results in the following sections.

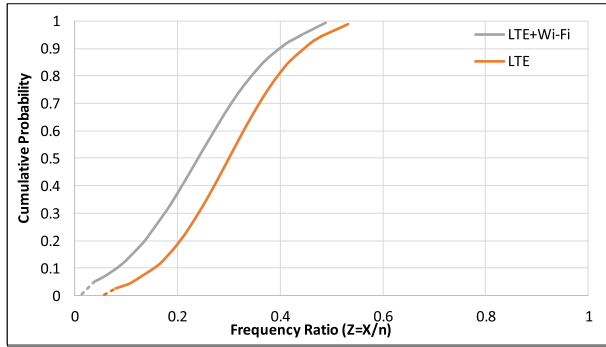


FIGURE 7. CDF of Frequency Ratio for LTE and Wi-Fi.

VII. PERFORMANCE ANALYSIS

This section provides a detailed analysis of the parameters discussed in Section VI to depict the significance of PAT. We used Samsung Galaxy S10 with Android Pie in the real-time environmental set-up to collect the data for the analysis.

A. FREQUENCY RATIO (Z)

As defined in Eq. (3), Z is a random variable defined as the frequency ratio of a domain name. As depicted in Eq. (4), the CDF of the collected data is aligned with the CDF curve of a normal distribution. The data is collected for a wide range of domain names from popular as well as region-specific applications with 100 trials to calculate frequency ratio for each domain name. The CDF plots for Z for the data collected using different network interfaces are shown in Fig 7. The distribution graph clearly shows that for more than 70% of the domain names, the frequency ratio lies below 0.35, which means that only 35% of the times, the application connects to the best server by default. Also, the maximum value of the frequency ratio is 0.48, which means that for 52% of the times, the application (for almost all the domain names) connects to the default server, which is not the best server at that moment. And, hence PAT can be useful in identifying the best server at that moment.

B. RTT RATIO (γ)

As discussed in Section VI, the difference in latency between default server and best server is captured using RTT ratio. It is calculated according to Eq. (5) for different domain names and plotted in Fig 8 as box plots. The box plots show the data collected over 100 trials for each domain name. γ = 1 means that the default server and the best server have same latency. The lesser the value of γ, the more is the variation of RTT of default server from best server. When the upper quartile of box for a domain name lies around γ = 0.7, it implies that for 75% of the trials the RTT of best server is atleast 30% better than the RTT of default server. Hence, migrating from default server to best server will significantly improve the latency.

C. RTT COMPARISON

Fig 9 shows the average RTT comparison of different mirror servers connected using both Wi-Fi and LTE. In Android

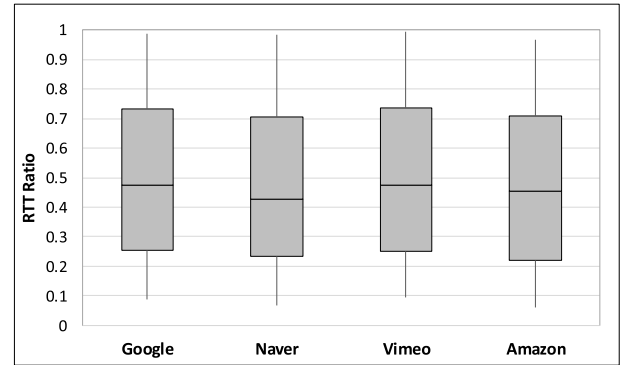


FIGURE 8. Analysis of RTT Ratio γ for Popular Applications.

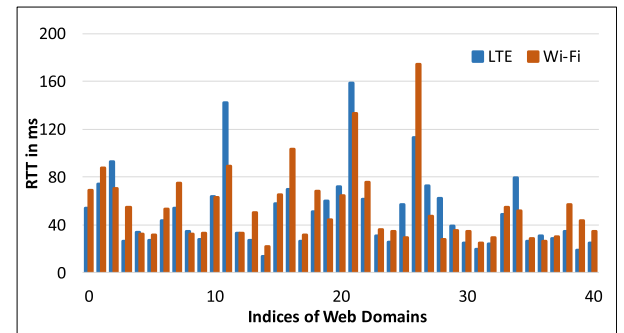


FIGURE 9. RTT Comparison for different servers with LTE and Wi-Fi. The results show that Wi-Fi is not the best choice for majority of the domains.

devices, Wi-Fi is preferred network over LTE even though low latency edge servers are available near to the mobile device. The result shows that in most of the cases, RTT of mirror server connected through Wi-Fi is comparatively more than the RTT of mirror server connected using LTE (or the edge server). Hence, PAT will be useful in identifying the edge servers and divert the connections from the preferred Wi-Fi network to LTE.

This performance analysis strongly suggests that the default server is not always the best. Hence, our proposed solution can play a significant role in resolving these shortcomings and will help in improving the connection throughput.

VIII. EXPERIMENTAL RESULTS

The proposed solution is evaluated in this section by comparing different network parameters for end to end TCP connections with and without PAT solution. The results for a few metrics are collected from simulated environment while for other metrics the results are from live air experiments, conducted locally, as discussed below.

A. PERFORMANCE EVALUATION ON SIMULATED ENVIRONMENT

We evaluated the performance of PAT for a 5G mmWave network using the ns3 simulation setup [30], as shown in Fig 10. The experimental setup includes both, 5G and Wi-Fi networks, with two separate mirror servers near to each network. The first mirror server (Mirror1) is deployed near to the

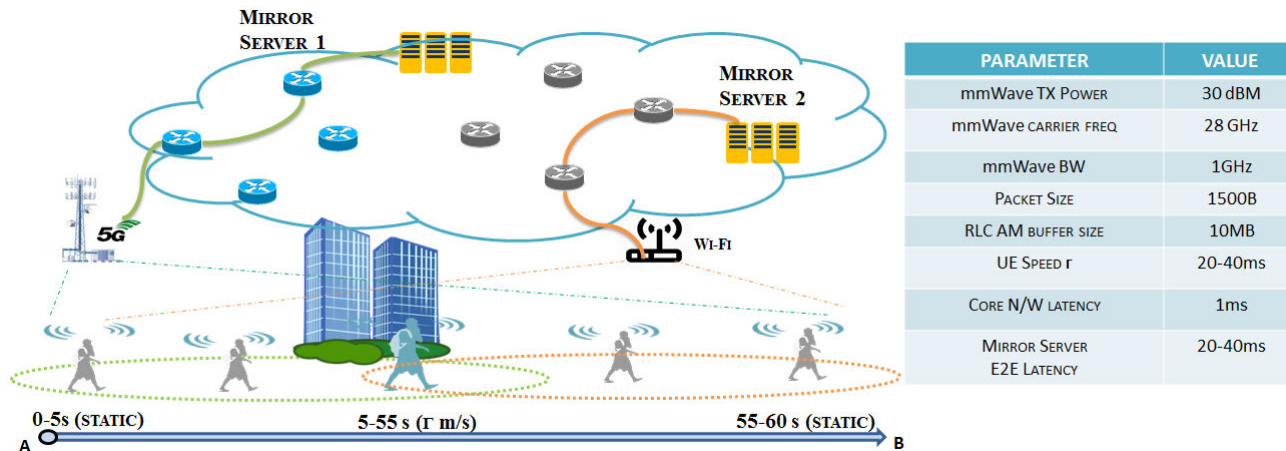


FIGURE 10. The performance of the end devices with/without PAT solution is evaluated in 5G and WiFi networks using the ns3 simulation setup with the configuration parameters as shown and the results are captured as the user moves from point A to point B.

5G mmWave network, and the other (Mirror2) is deployed close to the Wi-Fi back-haul. For evaluation, three devices are used simultaneously, where one of the devices (UE1) is configured with the PAT solution, and the other two devices (UE2 and UE3) are configured without our solution.

During the evaluation, UE1 is configured to connect with both the 5G and Wi-Fi networks. Whereas UE2 is always latched with 5G mmWave network, and UE3 is always connected to the Wi-Fi network. As depicted in Fig 10, all the three devices move from Point A (close to 5G network) to Point B (close to Wi-Fi backhaul) at a constant speed of 1.5m/s. There is a building 30 meters beside the path to simulate the blockage effect of mmWaves. UE2 and UE3 are always connected to Mirror1 and Mirror2 server, respectively, and hence the performance varies during user mobility. Whereas, UE1 using the PAT solution migrates the TCP connections between the mirror servers.

Next, to get a comprehensive analysis of the seamless migration, download time is analyzed using the same ns3 simulation environment. For that, we configured 5 HTTP mirror server instances with varied RTT. Each mirror server is deployed with replicated contents of different sizes (4MB, 16MB, 64MB, and 128 MB). PAT establishes simultaneous connections with these servers over available network interfaces and dynamically routes the data request to the estimated best server. Fig 11 shows the download time comparison of the contents of different sizes. As the RTT varies, PAT migrates the ongoing connections from 5G mmWave to Wi-Fi and vice-versa seamlessly. Hence, it downloads the content from multiple mirror servers and improves the download time from 6% to 33% consistently.

**B. LIVE AIR EXPERIMENT RESULTS**

We evaluated the performance of PAT using real-time analysis on Android applications. The PAT solution is implemented in Samsung flagship devices S10 and S10+ models. Popular and region-specific applications are installed on two identical devices with and without PAT solution. Both Wi-Fi and

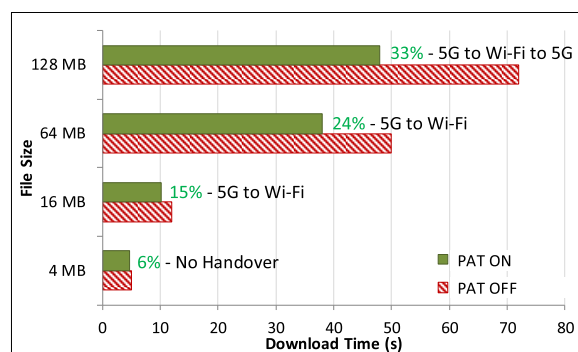


FIGURE 11. PAT significantly improves the download time from 6% to 33% as it enables seamless migration of the ongoing connections from 5G mmWave to Wi-Fi and vice-versa.

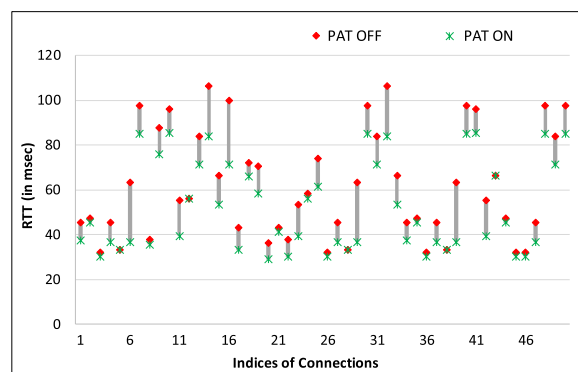


FIGURE 12. RTT Comparison over live air, with and without PAT, shows reduction in connection latency up to 15%.

Mobile data are enabled and the user is prompted to change the settings to allow switching between Wi-Fi and LTE.

In the first experiment, we used the End-To-End (E2E) latency as a performance evaluation parameter. Fig 12 shows the comparison of RTT for several TCP connections with and without the PAT solution. For each TCP connection request, PAT measures the RTT of all the available content servers and connects to the best server. From the results, we infer

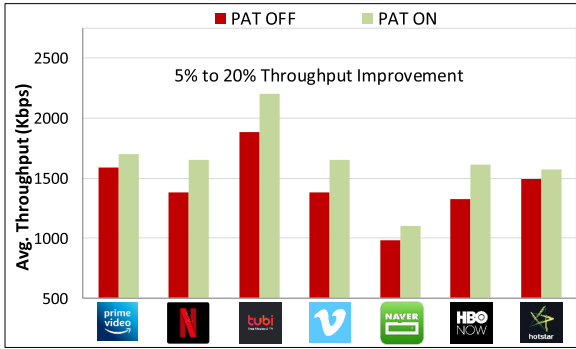


FIGURE 13. Average percentage gain in throughput( $\Delta_{tp}^{avg}$ ). The result shows an improvement in throughput by up to 20% with the PAT solution.

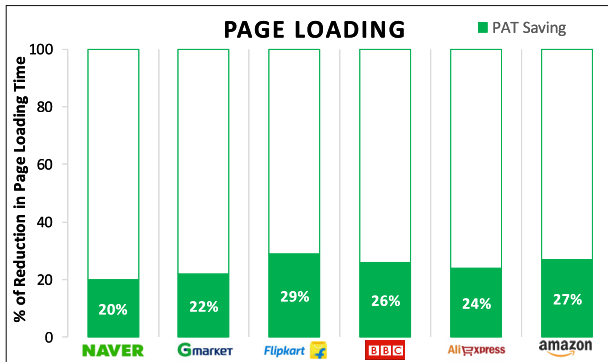


FIGURE 14. PAT reduces the Page Load Time consistently up-to 29%, which improves the user experience considerably.

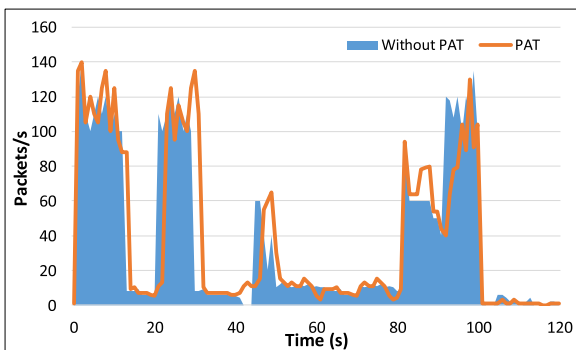


FIGURE 15. The packet I/O graph captured for the Naver application shows that there is no significant increase in the number of packets sent or received when PAT is enabled.

that PAT reduces the connection latency up-to 15%, which is a significant gain, especially for low-latency applications.

The second experiment was to evaluate the real time performance of PAT using popular video streaming applications. To ensure congruence results, we used different video streaming applications. We used percentage gain ( $\Delta_{RTT}^{avg}$ ) in throughput as mentioned in Eq. 10, as performance evaluation parameter in this experiment. The graph in Fig 13 represents the average throughput values collected from 50 trails. The result clearly shows an improvement of the connection throughput from 5% to 20% with the PAT solution.

Finally, we also conducted experiments to evaluate the performance of PAT for short-lived TCP connections.

We installed popular e-commerce, weather, and news applications. Page Loading Time (PLT) is used as an evaluation parameter in this experiment. As shown in Fig 14, PLT is reduced consistently up-to 29%, which improves the user experience considerably.

Since PAT creates parallel connections to find the best path, it is important to find out the excess network traffic created by PAT as compared to when PAT is disabled. The Fig 15 shows a two minute snapshot of the packet I/O graph captured for the Naver application. There is 0.88% increase in the number of packets sent or received when PAT is enabled. It is evident that there is no significant increase in the network traffic due to PAT.

### IX. CONCLUSION

This article presents the first-ever client only middleware solution called PAT, which explores all available content servers, finds the best available network path and dynamically migrates one TCP connection to another, seamlessly. This solution has been successfully prototyped in the Samsung Galaxy S10 device with Android P. The solution has been tested under diverse network conditions, and results show an improvement in content downloading time by 12% to 18% consistently. It also reduces page loading time drastically up to 29% when tested on popular Android apps.

In the future, we plan to integrate a learning module to PAT, which can estimate the best candidate paths for the end-to-end connection. The learning module can utilize past connection data and cross-layer information for the estimation. It would further improve the performance of PAT as the Path Navigator can then connect and monitor only these selective few candidate paths.

### REFERENCES

- [1] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May/Jun. 2020.
- [2] Z. Pi and F. Khan, "An introduction to millimeter-wave mobile broadband systems," *IEEE Commun. Mag.*, vol. 49, no. 6, pp. 101–107, Jun. 2011.
- [3] V. Raghavan, L. Akhoondzadeh-Asl, V. Podshivalov, J. Hulten, M. A. Tassoudji, O. H. Koymen, A. Sampath, and J. Li, "Statistical blockage modeling and robustness of beamforming in millimeter-wave systems," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 7, pp. 3010–3024, Jul. 2019.
- [4] J. Lu, D. Steinbach, P. Cabrol, and P. Pietraski, "Modeling human blockers in millimeter wave radio links," *ZTE Commun. Mag.*, vol. 2012, pp. 23–28, Dec. 2012.
- [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. Commun., 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, 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.
- [7] T. Callahan, M. Allman, and M. Rabinovich, "On modern DNS behavior and properties," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 3, pp. 7–15, Jul. 2013.
- [8] 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.



- [9] 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.
- [10] J. Kim, Y.-C. Chen, R. Khalili, D. Towsley, and A. Feldmann, "Multi-source multipath HTTP (mHTTP) a proposal," in *Proc. ACM Int. Conf. Meas. Modeling Comput. Syst. (SIGMETRICS)*, New York, NY, USA, 2014, pp. 583–584.
- [11] Y.-C. Chen, D. Towsley, and R. Khalili, "MSPlayer: Multi-source and multi-path video streaming," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 8, pp. 2198–2206, Aug. 2016.
- [12] F. Ulupinar, P. Purkayastha, R. Walker, Y. Mao, and B. H. Kelley, "Transport accelerator systems and methods," U.S. Patent 9930097 B2, Jul. 3, 2014.
- [13] A. Vulimiri, P. B. Godfrey, R. Mittal, J. Sherry, S. Ratnasamy, and S. Shenker, "Low latency via redundancy," in *Proc. 9th ACM Conf. Emerg. Netw. Exp. Technol.*, 2013, pp. 283–294.
- [14] K. Park, V. Pai, L. Peterson, and Z. Wang, "CoDNS: Improving DNS performance and reliability via cooperative lookups," in *Proc. 6th Symp. Oper. Syst. Design Implement.*, Jan. 2004, pp. 199–214.
- [15] U. Goel, M. P. Wittie, and M. Steiner, "Faster Web through client-assisted CDN server selection," in *Proc. 24th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2015, pp. 1–10.
- [16] 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.
- [17] 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.
- [18] J. Iyengar, C. Raiciu, S. Barre, M. J. Handley, and A. Ford, *Architectural Guidelines for Multipath TCP Development*, document RFC 6182, Mar. 2011. [Online]. Available: <https://rfc-editor.org/rfc/rfc6182.txt>
- [19] J. Wu, C. Yuen, B. Cheng, M. Wang, and J. Chen, "Streaming high-quality mobile video with multipath TCP in heterogeneous wireless networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 9, pp. 2345–2361, Sep. 2016.
- [20] A. Nikraves, Y. Guo, F. Qian, Z. M. Mao, and S. Sen, "An in-depth understanding of multipath TCP on mobile devices: Measurement and system design," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, Oct. 2016, pp. 189–201.
- [21] O. Levillain, "A study of the TLS ecosystem," Ph.D. dissertation, Institut Nat. des Télécommun., Ghaziabad, India, 2016.
- [22] K. Arunachalam, J. M. Ppallan, K. Govindan, S. Jaiswal, K. Subramaniam, and V. Balasubramanian, "Layer 4 optimizer (L4O) for enhancing battery life in smart devices," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–7.
- [23] 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.
- [24] K. Govindan, A. Karthikeyan, J. M. Ppallan, S. Jaiswal, and K. Subramaniam, "TCP closure optimization for enhanced battery life in smart devices," *IEEE Trans. Mobile Comput.*, vol. 18, no. 3, pp. 645–657, Mar. 2019.
- [25] J. M. Ppallan, K. Arunachalam, K. Govindan, S. Jaiswal, and K. Subramaniam, "Optimizing TCP zero window probes for power saving in smart devices," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2018, pp. 1–6.
- [26] Wikipedia Contributors. (2018). *Central Limit Theorem—Wikipedia, the Free Encyclopedia*. Accessed: Apr. 14, 2019. [Online]. Available: <https://en.wikipedia.org/wiki/Centrallimittheorem>
- [27] A. Gurtov, T. Henderson, S. Floyd, and Y. Nishida, *The New Reno Modification to TCP's Fast Recovery Algorithm*, document RFC 6582, Apr. 2012. [Online]. Available: <https://rfc-editor.org/rfc/rfc6582.txt>
- [28] S. Ha, I. Rhee, and L. Xu, "Cubic: A new TCP-friendly high-speed TCP variant," *SIGOPS Oper. Syst. Rev.*, vol. 42, no. 5, pp. 64–74, Jul. 2008, doi: [10.1145/1400097.1400105](https://doi.org/10.1145/1400097.1400105).
- [29] N. Wu and A. Tang, "End-to-end network throughput optimization through last-mile diversity," in *Proc. 52nd Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2018, pp. 1–6.
- [30] R. Ford, M. Zhang, S. Dutta, M. Mezzavilla, S. Rangan, and M. Zorzi, "A framework for end-to-end evaluation of 5G mmWave cellular networks in ns-3," in *Proc. Workshop ns-3 (WNS)*, 2016, pp. 85–92.



**SHIVA SOUHITH GANTHA** (Member, IEEE) received the B.Tech. degree in electrical engineering with minors in computer science and engineering from IIT Bombay, Mumbai, India, in 2019. He is currently working as an Engineer with the Samsung Research and Development Institute, Bengaluru, India. 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 the 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 the Samsung Research and Development Institute, Bengaluru, India. Previously, she worked as a Software Engineer with Tata Consultancy Service

Ltd. Her research interests include communication and networking which include next-generation transport layer protocols, multi-access edge computing (MEC), 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 Research and Development Institute, Bengaluru, India. Previously, he worked as a Senior Software Engineer with Huawei Technologies India Private

Limited and an Associate Programmer with the National Informatics Centre, India. His research interests include next-generation transport and network layer protocols, cross-layer optimization, smartphone operating systems, and green communications.



**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 the Samsung Research and Development Institute, Bengaluru, India. Previously, he worked as a Senior Associate with Novell Software Development India Private Limited, a Senior Software

Engineer with Huawei Technologies India Private Limited, and a Software Engineer with Protechsoft Technologies Private Limited. His current research interests include next-generation transport layer protocols, cross-layer communication, and mobile edge computing. He is a member of ACM.

...