International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 16, No.1 (Jul-24)

http://dx.doi.org/10.12785/ijcds/160106

Enhanced Multipath TCP to Improve the Mobile Device Network Resiliency

Hilal H. Nuha[1], **Fazmah Arif Y.**[1] **and Hendrawan**[2]

[1]*School of Computing, Telkom University, Bandung, Indonesia*
[2]*School of Electrical Engineering and Informatics, Institute of Technology Bandung, Bandung, Indonesia*

Received 19 May. 2023, Revised 22 Mar. 2024, Accepted 6 Apr. 2024, Published 1 Jul. 2024

Abstract: Mobile devices consume a significant amount of internet traffic, and they can utilize multiple interfaces like Wi-Fi and cellular networks to share traffic between networks, known as a multi-homed host. This approach enhances the resilience of the internet connection by allowing traffic to flow through multiple paths. The Multipath Transmission Control Protocol (MPTCP) supports this type of connection, but a fairness issue emerges when a multi-path host shares a bottleneck link with regular single-path hosts. To deal with this issue, this paper proposes an enhanced MPTCP (eMPTCP) that uses a throughput adjustment to estimate bandwidth based on TCP Westwood+ congestion control. By decreasing each subflow traffic on the multi-path, the proposed eMPTCP achieves fairness in shared links. The simulation conducted using network simulator 2 (ns2) represents the mobile conditions of mobile data air interface, and the results demonstrate that eMPTCP outperforms standard congestion control in achieving connection resilience.

Keywords: Multi-path TCP, Fairness, Congestion Control, Bandwidth Estimation

1. Introduction

Smartphones currently account for a significant proportion of internet traffic, estimated to be around 57.84% globally in March 2022 [1], [2]. Most smartphones are multi-homed devices, as they typically have two or more network interfaces (NIs) [3], each with its own unique address. Furthermore, cellular traffic is expected to grow at a rate that is ten times faster than fixed internet traffic, highlighting the importance of optimizing for multi-homed devices [4] to improve the connection resilience. This enables the device to make use of multiple networks simultaneously, a feature facilitated by Multipath Transmission Control Protocol (MPTCP) [5], [6]. However, when MPTCP competes with normal single interface hosts in a shared bottleneck link, a fairness issue arises.

Several studies and proposals on MPTCP have been reported in literature. Chao et al. [7] discuss the importance of Multipath TCP (MPTCP) as an extension to TCP that allows the use of multiple paths in data transmissions, offering bandwidth utilization, fairness, and resilience, and highlights its potential in solving challenges faced by vehicular Internet-of-Things systems, while also identifying future research directions. Li et al. [8] develop a learning-based multipath congestion control approach called SmartCC to handle the diverse communication paths in heterogeneous networks and shows that it increases the aggregate throughput significantly and achieves the best performance metrics among different methods. Chaturvedi et al. [9] propose a new adaptive and efficient packet scheduler (AEPS) that overcomes performance degradation and achieves high throughput with least completion time by exploiting the bandwidth of all available paths, delivering data packets in-order to the receiver, and outperforming existing schedulers. It can be noticed that one of the challenges of MPTCP is to achieve fairness among MPTCP flows and between MPTCP and regular TCP flows. Several studies have investigated different aspects of MPTCP fairness [10], such as congestion control algorithms[11], [12], router support, overlay networks, and application-level relay. These studies have proposed various methods to improve MPTCP fairness, such as tuning parameters [13], [14], modifying TCP mechanisms, exploiting multipath characteristics, and using relay nodes. However, there are still open issues and trade-offs in MPTCP fairness, such as stability [15], scalability, heterogeneity, and overhead. Fang et al. [16] evaluate sleep-scheduling policies in multiple access networks. Hou et al. [17] integrate fixed-edge and mobile-edge computing nodes

Motivated by the work in [18], on using Bidimensional-Probe Multipath Congestion Control (BMC) as a combination of several subflows in a multipath connection fairly utilizes with background TCP-friendly flows under bottleneck link, this paper proposes a congestion control mechanism by scaling down each subflow of the MPTCP. Each congestion window size is governed by the weight calculation obtained from the bandwidth estimation mechanism utilized by TCP Westwood+ congestion control [19]. This multipath protocol is further referred to as the enhanced MPTCP (eMPTCP) that is expected to improve the connection resilience without damaging the fairness.

The remainder of this paper is structured as follows. Methodologies are discussed in section II. Proposed method is provided in section III. Experimental results and discussion are given in section IV. Finally, section V summarizes this work by conclusions.

2. Methodologies

In this section, the basis of the TCP congestion control to avoid congestive collapse consisting a multi-faceted congestion-control strategy [20] is provided. First, four congestion control algorithms are presented in order, namely, slow start (SS), congestion avoidance (CA), fast retransmit, and fast recovery. Second, the concept of multipath for TCP is presented. Lastly, the proposed MPTCP congestion control is introduced.

A. Congestion Control

During SS phase, the W or cwnd value is governed by the following relation:

$$W \leftarrow W + 1 \ per \ ACK \tag{1}$$

or

$$W \leftarrow 2W \ per \ RTT \tag{2}$$

And

$$W \leftarrow W/2 \ per \ Loss \tag{3}$$

The particular size is called as sstresh or slowstart threshold [21]. When *cwnd* exceeds the size of sstresh, the system starts the CA phase with different rules:

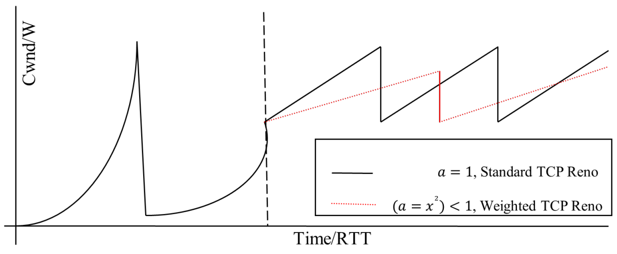

Figure 1. cwnd size on TCP.

Fig. 1 shows the size of W in SS and CA phases where the W is governed by the following rules:

Additive Increase

$$ACK \rightarrow W \leftarrow W + \frac{1}{W} \tag{4}$$

Multiplicative Decrease

$$LOSS \rightarrow W/2 \tag{5}$$

where $1/W$ represents the increment coefficient. TCP Reno (TCPR) CA phase follows Additive Increase Multiplicative Decrease (AIMD).

According to the research conducted in [18], TCPR's throughput (T^r) can be expressed as a function of packet size (s), packet loss rate (p), timeout value (t), and RTT value (R).

$$T^r = \frac{s}{R\sqrt{\frac{2p}{3a}} + t(3\sqrt{\frac{3p}{8a}})p(1+32p^2)} \tag{6}$$

$$T^r = \sqrt{a}\left(\frac{s}{R\sqrt{\frac{2p}{3a}} + t(3\sqrt{\frac{3p}{8a}})p(1+32p^2)}\right) \tag{7}$$

$$T^r_{a=1} = \left(\frac{s}{R\sqrt{\frac{2p}{3}} + t(3\sqrt{\frac{3p}{8}})p(1+32p^2)}\right) \tag{8}$$

The equation mentioned implies that the throughput of TCPR is calculated by taking the square root of an increment coefficient denoted as a. In regular TCP, the value of a is typically set to 1. However, if a higher throughput is desired, the value of a can be increased to x^2, where x represents the factor by which the normal throughput is expected to be multiplied.

$$Th^r_{a=x^2} = \sqrt{x^2}\left(\frac{s}{R\sqrt{\frac{2p}{3a}} + t(3\sqrt{\frac{3p}{8a}})p(1+32p^2)}\right) \tag{9}$$

$$Th^r_{a=x^2} = xTh^r_{a=1} = x\left(\frac{s}{R\sqrt{\frac{2p}{3a}} + t(3\sqrt{\frac{3p}{8a}})p(1+32p^2)}\right) \tag{10}$$

where a depends on the TCP type. For example, TCPR uses $a = 1$ and Th is the throughput achieved by using the default parameter, according to BMC in [18], if a is set to $x^2 = (\frac{1}{2})^2 = \frac{1}{4}$, the protocol achieves $\frac{1}{2}$ Th or half of the default throughput. This indicates that for $a = x^2$, the achieved throughput is x times of Th where $0 < x^2 1$.

B. TCP Westwood+

It can be noticed that link bandwidth estimation is crucial for MPTCP to determine the proper weight of each subflow. Therefore, this paper utilize the bandwidth estimation provided by the TCP Westwood+ (TCPW+) which is an enhancement of TCP Westwood (TCPW) for networks long delay and high loss rate like cellular and satellite system [22]. In linux kernel 2.6, TCPW+ is a built in protocol and can be used after some configurations.

TCPW+ uses similar SS algorithm with TCPR:

$$W \leftarrow W + 1 \ per \ ACK \tag{11}$$

$$W \leftarrow 2W \ per \ RTT \tag{12}$$

$$W \leftarrow W/2 \ per \ Loss \tag{13}$$

The SS algorithm in TCP enables the window size (W) to increase exponentially and quickly until it reaches a threshold value (ssthresh). At this point, TCP enters the CA phase. The primary difference between TCPR and TCPW+ is in their CA algorithm, which falls under the category of Additive Increase Adaptive Decrease (AIAD). This is because TCPW+ adapts differently than TCPR in response to loss conditions, as mentioned in [22].

$$W = \begin{cases} W + \frac{a}{W}, \ a = 1, \ ACK, & additive \ increase \\ max\left(2, \ \frac{BWE*RTTmin}{Seg_{size}}\right), & adaptive \ decrease \end{cases} \tag{14}$$

In reference to [23], the steady-state throughput of TCPW+ is determined by three variables: available bandwidth end-to-end (BWE), the size of the TCP segment in bits ($Segsize$), and the minimum round trip time measured during the connection (RTT_{min}) as denoted by the following equation.

$$Th^{west} = \frac{1}{\sqrt{R.(R - R_{min})}} \left(\sqrt{\frac{a(1-p)}{p}} \right) \tag{15}$$

$$Th^{west} = \sqrt{a} \left(\frac{1}{\sqrt{R.(R - R_{min})}} \sqrt{\frac{a(1-p)}{p}} \right) \tag{16}$$

$$t_q = (R - R_{min}) \tag{17}$$

$$Th^{West+}_{a=1} = \left(\frac{1}{\sqrt{R.t_q}} \sqrt{\frac{a(1-p)}{p}} \right) \tag{18}$$

The equation above shows that the throughput of TCPW+ is directly proportional to the square root of the increment coefficient denoted by a. In comparison, standard TCP uses an increment coefficient of 1. If a higher throughput is desired, the value of a can be increased to x^2, where x is the factor by which the regular throughput is expected to be multiplied.

One of the primary differences between the throughputs of TCPR and Westwood+ is in their dependence on the RTT. The throughput of TCPR is inversely proportional to the RTT, whereas the throughput of TCPW+ is inversely proportional to the square root of the RTT.

$$Th^{west+}_{a=x^2} = \sqrt{x^2} \left(\frac{1}{\sqrt{R.t_q}} \sqrt{\frac{a(1-p)}{p}} \right) \tag{19}$$

$$West+ = x.Th^{west}_{a=1} \tag{20}$$

Therefore, the algorithm is transformed into following formula:

$$W = \begin{cases} W + \frac{x^2}{W}, per \ ACK & additive \ increase \\ max\left(2, \frac{BWE*RTTmin}{Seg_{size}}\right), per \ LOSS & adaptive \ decrease \end{cases} \tag{21}$$

Therefore, the alterations also impact the achieved throughput within the network.

C. Multihomed Device and Multipath TCP

A host with multiple TCP connections is defined as a multipath host. From Fig. 2 , UE1 is depicted as a mobile device with three TCP connections from a host and UE2 as a normal single connection mobile device.

Because UE1 has three connection, the uplink throughput dominates the networks as depicted by following figure:

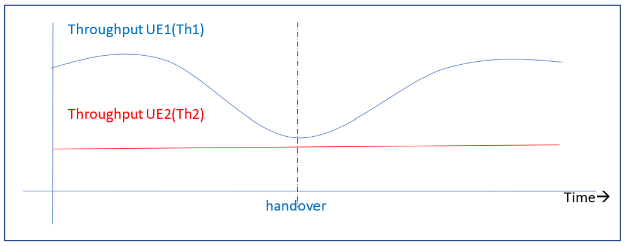

Figure 2. Multipath Mobile Device

On an underutilized network, the throughput of UE2 will not be disturbed by UE1, but on a bottleneck condition, the UE2 will only get $\frac{1}{4}$ of total bandwidth. This conditions violates the fairness.

D. Max – Min Fairness

In this paper, the main assumption applied to the resource utilization is the Max-Min fairness rule [22] that can be achieved by the following steps:

1) First, all flows throughput are set to zero.
2) Each flow throughput grows at the same rate until it reaches the capacity of one or more link on the path corresponding to the flow.
3) The remaining flows throughput keep growing.
4) Step 2 is repeated until all flows are unable to increase the throughput.

E. TCP Modifications for Achieving Fairness

According to the discussion above, the TCP is modified to achieve fairness and additional schemes are added on the higher layer [18]. Each subflow throughput (Th_n) is reduced in proportion to the weight value. The scheme is applied to the transport layer which has direct access to the TCP parameter. The scheme uses an increment coefficient a with a coefficient $D_n^2 = x^2$ to reduce the aggressiveness. Therefore, the sum of each subflow throughput of the multipath host is equal to a standard TCP flow. As depicted by Fig. 3, the unfair bottleneck problem is described as follows:

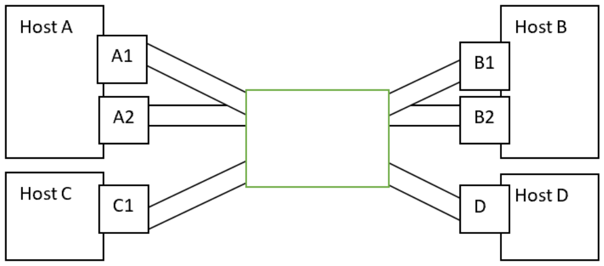

Figure 3. Multipath Mobile Device

Consider the bandwidth b represents the maximum capacity of the bottleneck link and each subflow maximum capacity is also b, then the total throughput achieved by the multipath host with N subflow is given by:

$$Th_{MP} = Th_1 + Th_2 + ... + Th_N = Th + Th .. + Th = NTh \quad (22)$$

Where Th_{MP} denotes the multipath TCP throughput. However, the throughput achieved by the standard TCP is given by:

$$Th_s = Th \quad (23)$$

This weight managing scheme must find the coefficient that is able to regulate the throughput. Consider the increment coefficient (a) of each MPTCP subflow is given by :

$$a = x^2 \quad (24)$$

Let the sublow increment coefficient x_n^2 is given by the following equation:

$$x_n^2 = \frac{BW_n}{BW_1 + .. + BW_N} \quad (25)$$

where BW_N is the bandwidth of the link n corresponding to the subflow n and $0 < x < 1$.

This step is also included in a process that determines the weight of each connection. By replacing the increment coefficient of each subflow in the multipath host with x^2:

$$x = \frac{BW_n}{BW_1 + BW_2 + .. + BW_N} \quad (26)$$

$$x = \frac{1}{n} \quad (27)$$

Thus, the bottleneck condition becomes.

$$Th_{mul} = T_1 + T_2 + ... + T_N = \frac{1}{n}T' + \frac{1}{n}T' .. + \frac{1}{n}T' = \frac{n}{n}T' = T' \quad (28)$$

The throughput is also applicable to a single path as follows:

$$T_{single} = T' \quad (29)$$

when $T_{single} = T' > T$, it implies that a single path is in fair competition with a multipath connection. As described earlier, TCPW+ includes a parameter called $bwe\_$, which indicates the estimated bandwidth of a connection. The method for managing the weights considers the bandwidth estimation of each connection ($bwe\_$) [6] and calculates the value of x for each connection. This value of x is then returned to the corresponding connection as its increment coefficient, which is set to x^2. The calculation for x is as follows:

$$x = \frac{bwe_{max}}{bwe_1 + bwe_2 + .. + bwe_n}, \quad (30)$$

The reason for using Equation (30) instead of Equation (27) is because it is more suitable for mobile networks which experience signal fading and thus have varying bandwidths. Equation (30) requires knowledge of the maximum bandwidth estimation (BWE_{max}) from a single path. In the simulation, different values are used to substitute the maximum estimated bandwidth BWE_{max}. $bwe_1...bwe_n$ represent the estimated bandwidth estimation of each subflow on a multipath host, and the equation 30 updates the weight value x periodically every t seconds.

F. *Multipath TCP Architecture and Implementation*

MPTCP is a protocol that allows hosts with multiple addresses to utilize multiple paths to connect to other hosts, enhancing throughput and connection resilience. The protocol distributes traffic among available paths, responds to congestion effectively, and optimizes overall network utility by reducing congestion and utilizing spare capacity. MPTCP uses standard TCP sessions, called "subflows," which are compatible with the network, and MPTCP-specific information is carried separately from the actual data being transferred. To use MPTCP, at least two available paths are required between hosts, regardless of the number of addresses on each host, and shared bottlenecks can be handled by the MPTCP congestion controller [24].

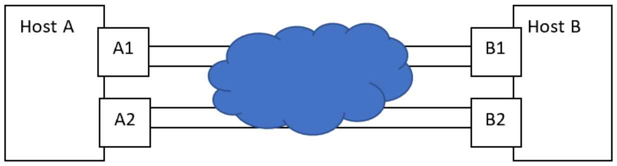

Figure 4. Multipath Mobile Device

MPTCP has gained attention for its ability to improve network utilization and provide higher resilience through the use of multiple paths. However, to achieve optimal performance, MPTCP must address several challenges, including congestion control. To address this, Modified Congestion Control (MCC) was proposed to ensure that each subflow has its own congestion control state, allowing capacity on each path to be matched by offered load. While running TCP New Reno on each subflow may be a simple solution, it can result in an unfair share of the network's resources when the subflows share a common bottleneck.

To ensure fair and efficient use of network resources, a practical multipath congestion control algorithm should meet three key goals:

1) Improve throughput by performing at least as well as a single path flow on the best available path.
2) Do no harm by not taking up more capacity on any one path than a single path flow would.
3) Ensure bottleneck fairness by avoiding unfair resource allocation when subflows share a common bottleneck. By meeting these goals, MPTCP can achieve optimal performance and provide better resilience for network applications.

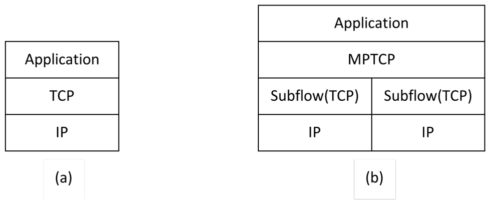

Figure 5. Comparison of Standard TCP and MPTCP Protocol Stacks.

MPTCP's layered architecture consists of subflow interface, path management, congestion control, and packet scheduling. Path management discovers and initializes multiple paths between hosts, while the packet scheduler segments data into connection-level segments, adds a sequence number, and sends it to a subflow. Congestion control schedules packets to be sent on subflows to avoid taking more bandwidth than a single-path TCP flow at a shared bottleneck. These functions work together to optimize overall network utilization by transferring load away from congested bottlenecks and taking advantage of spare capacity [25]. Figure 5 shows the layered architecture of MPTCP.

3. Proposed Method

A. *Algorithm of Enhanced Multipath TCP Based on Westwood+*

For each connection subflow, the following algorithm in Figure 6 is implemented.

This algorithm works on each TCPW+ connection after each connection reaches CA phase. In the beginning of connections, the connections behave like normal connection, once the connections reach the CA phase the Weight Managing calculates the bandwidth and find the x or the weight of each value. After x is calculated, the throughput adjustment will substitute increment coefficient a with x^2. That cycle is repeated after t second, so that the weight value is always updated. This algorithm is further referred to as enhanced Multipath TCP or simply eMPTCP. In addition to the throughput adjustment method, this also develops the weight managing method to calculate the correct weight for each subflow. The following figure 7 describes the comparison between standard TCP (Fig. 7(a)) and the eMPTCP (Fig. 7(b)).

According to throughput adjustment as defined by eq. 31, for each subflow the congestion control is modified into:

$$W = \begin{cases} W + x^2/W, \ a = 1, & ACK, additive\ increase, x^2 \\ max\left(2, \frac{BWE*RTTmin}{Seg_{size}}\right), & LOSS, adaptive\ decrease \end{cases} \tag{31}$$

The x value itself follows the weight managing method as defined by eq. 32:

$$x = \frac{bwemax}{bwe_1 + bwe_2 + ...bwe_n} \tag{32}$$

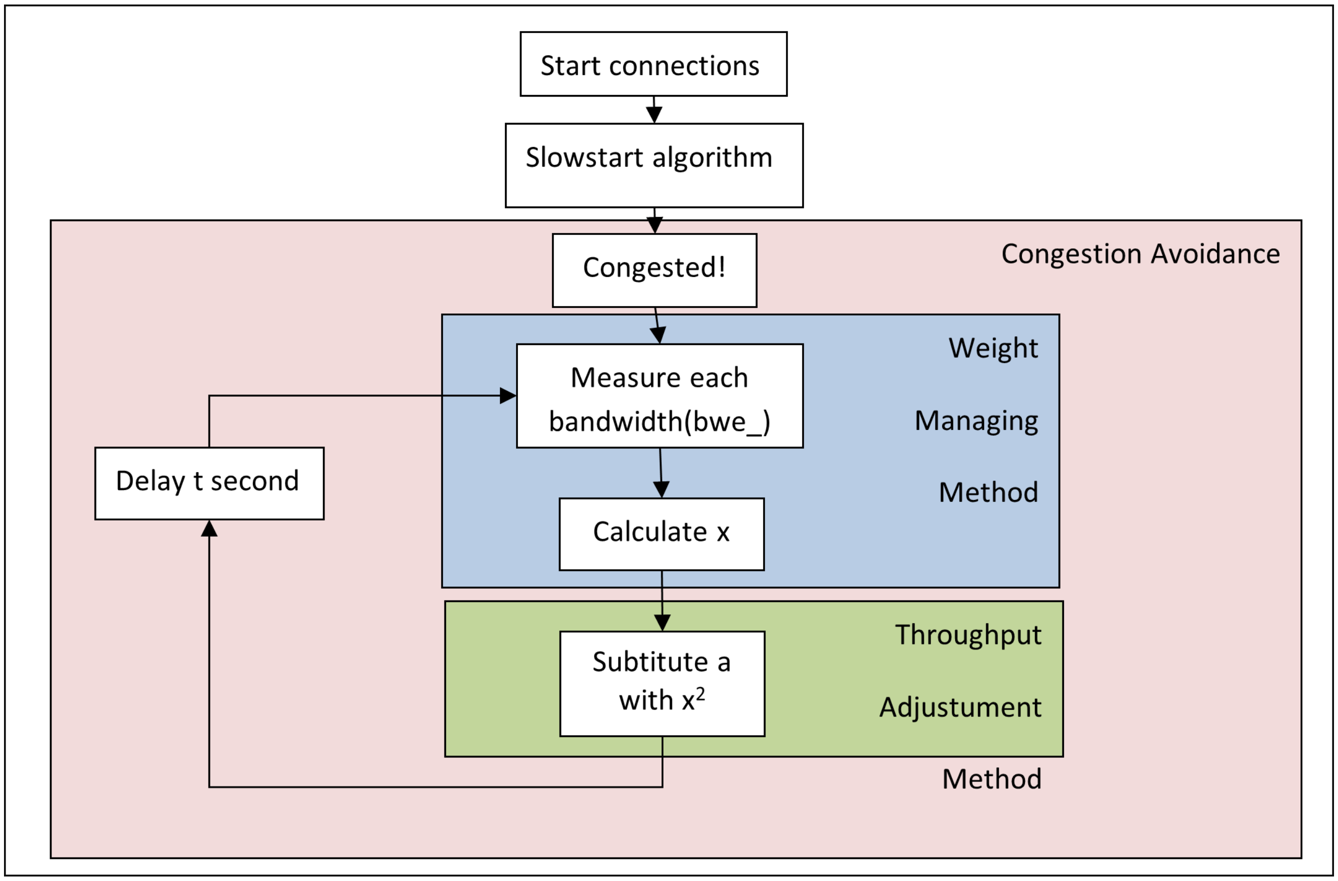

Figure 6. The Developed eMPTCP

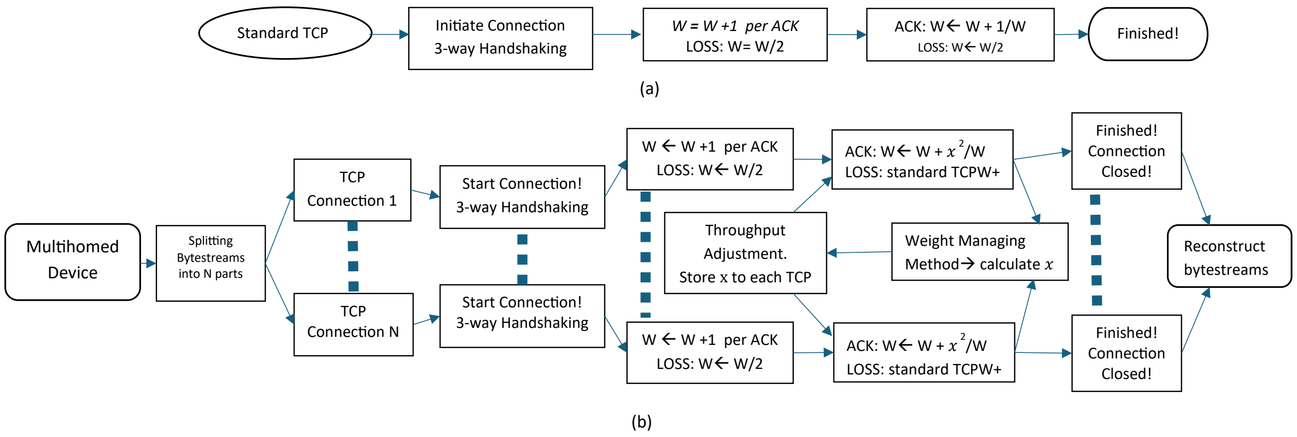
Standard TCP
Initiate Connection 3-way Handshaking
W = W +1 per ACK LOSS: W= W/2
ACK: W← W + 1/W LOSS: W← W/2
Finished!
(a)
Multihomed Device
Splitting Bytestreams into N parts
TCP Connection 1
TCP Connection N
Start Connection! 3-way Handshaking
W ← W +1 per ACK LOSS: W← W/2
Throughput Adjustment. Store x to each TCP
ACK: W← W + x²/W LOSS: standard TCPW+
Weight Managing Method→ calculate x
Finished! Connection Closed!
Reconstruct bytestreams
(b)

Figure 7. (a) Standard TCP and (b) Proposed eMPTCP

Using x^2 as increase factor will slow down the growth rate of throughput for each subflow.

Thus, once the limit throughput on bottleneck is reached the total throughput of each subflow is further scaled down and fairness on bottleneck condition can be reached.

B. Experiment Design

The study aimed to assess the fairness of a shared bottleneck when one of the hosts had a multipath connection. Network Simulator 2 with the EURANE patch was used to simulate a mobile environment, particularly during blackout periods that have very high loss conditions due to handover occurrences. The experiment involved two hosts located a long distance from the base station, with one being a multipath host located 2 km away. The total delay from the base station to the destination was 29 ms, and the bandwidth was fluctuating due to the signal condition, which varied depending on the distance as the simulation was carried out on a mobile host moving at 20 m/s towards a base station. Two experiment scenarios were performed to evaluate the fairness and throughput adjustment of the throughput of a host and the multipath host, respectively

1) Throughput Adjustment Evaluation

In this paper, two major experiment scenarios were conducted to evaluate the algorithm. The first one was to evaluate the throughput adjustment which tested whether the weight of a connection influences the throughput. As depicted by Figure 8, Host 1 is weighted by D_n and the experiment evaluates if the throughput of host 1 (T_1) compared to the throughput of host 2 (T_2) equals $T_1 : T_2 = D_n : 1$.

2) Weight Managing Evaluation

The second experiment evaluated if the Weight Managing would make the total multiple connection throughput of host 1 (T_1) competed fairly with other single connection Hosts (Host 2 and Host 3, T_2 and T_3, respectively). To check the stability, host 1 moved approaching each Base Station so each signal and bandwidth changed periodically. This experiment is depicted in Figure 8.

C. Analyzing Methods

To analyze relation between signal strength fluctuation and ratio between multipath throughput and normal single path, Network Simulator 2 (ns2) is employed using EURANE patch.

From the previous descriptions, we compared the total throughput from host 1. The result of the first experiment is analyzed based on its throughput ratio with respect to another host throughput. According to the hypotheses, the ratio becomes $T_1 : T_2 = D_n : 1$ since Host 1 is weighted by D_n.

The second experiment, Host 1 activates all its connections and the weight Managing makes each connection proportionally weighted. So the total throughput of the multipath connection compared to other single connection hosts (Host 2,3) will equal to $T_1 : T_2 : T_3 = 1 : 1 : 1$. There are four categories based on the ratios :

1) $T_1/T_{2,3}| < 1$ means the algorithm does not work optimally, since the multipath host cannot compete against other connection ($T_1 < T_{2,3}$).
2) $|T_1/T_{2,3}| = 1$ means the algorithm works optimally and successfully manages the proportion $T_1 = T_2 = T_3$.
3) $1 < |T_1/T_{2,3}| < 3$ means the algorithm also does not work optimally since the multipath dominate the shared bottleneck ($T_1 > T_{2,3}$) but the domination is less than without weight.
4) $|T_1/T_{2,3}|3$ means the algorithm does not work since the domination is not decreasing.

4. Results and Discusssion

Following the previous section, a series of experiments were carried out to assess the effectiveness of the proposed eMPTCP. Two methods, namely Throughput Adjusment and Weight Managing, were evaluated. The first method is addressed in the initial part of this section, while the latter is examined in the latter part of this section.

A. throughput adjustment Evaluation

To evaluate the capability of throughput adjustment to govern the throughput, the proposed method was evaluated using topology in Figure 9 :

The experiment configuration involves several components in the Universal Mobile Telecommunications System (UMTS). The User Equipment (UE) is any device used by an end-user to communicate and connects to the base station ("Node B") via the Uu radio interface. The Node B contains radio transmitters and receivers that communicate directly with mobile devices, while the Radio Network Controller (RNC) controls the Node Bs, carries out radio resource management, and manages some of the mobility functions. The Serving GPRS Support Node (SGSN) delivers data packets from and to mobile stations within its geographical service area, while the Gateway GPRS Support Node (GGSN) interworks between the GPRS network and external packet-switched networks. Finally, the Destination (D) is the end host that receives the packet from the UE.

The aforementioned network components use parameters given by Table I:

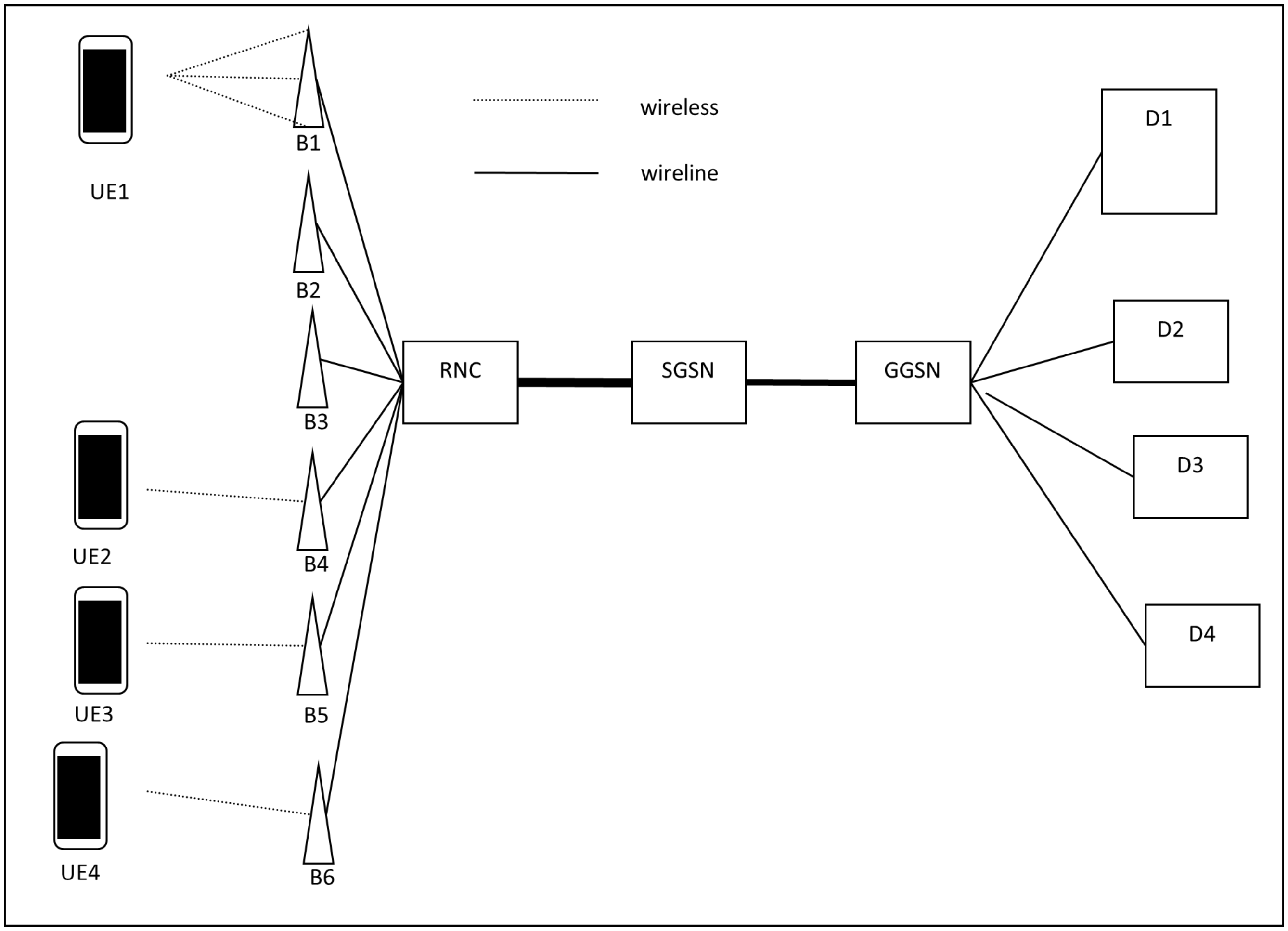

Figure 8. Experiment Design to Evaluate the Throughput Scaling

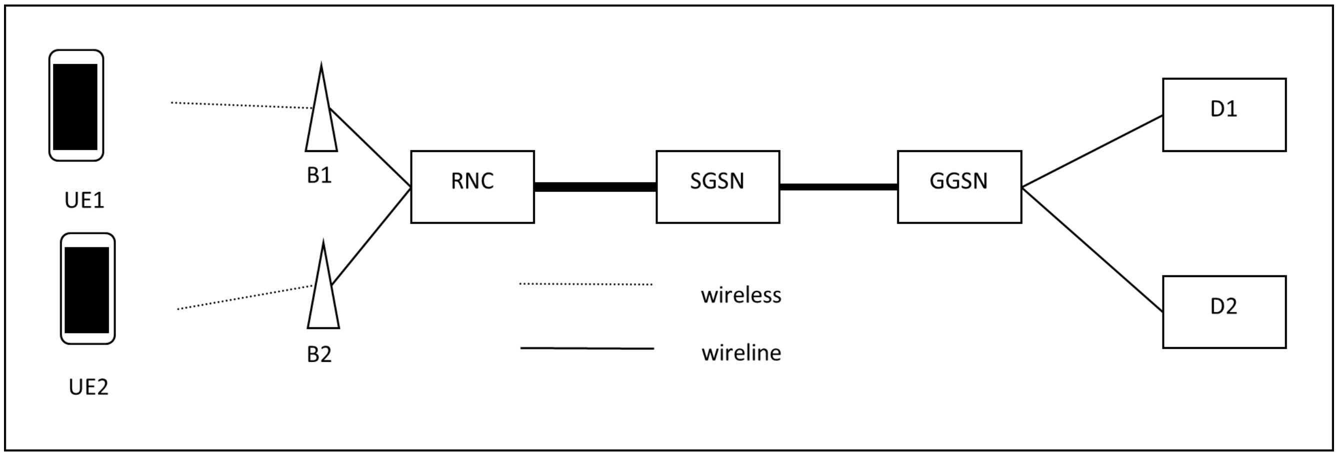

Figure 9. Throughput Scaling Experiment Scenario

TABLE I. Network Parameters for throughput adjustment Evaluation

no	Parameters	Value
1.	BS Downlink BW	32kbps
2.	BS Downlink TTI	10ms
3.	BS Uplink BW	32kbps
4.	BS Uplink TTI	10ms
5.	BS-RNC(Iub) Uplink/Downlink BW	155Mbps
6.	BS-RNC(Iub) Uplink/Downlink Delay	15ms
7.	RNC-SGSN Duplex BW(Bottleneck)	0.1Mbps
8.	RNC-SGSN Duplex Delay	6ms
9.	GGSN-SGSN Duplex BW(Bottleneck)	0.1Mbps
10.	GGSN-SGSN Duplex Delay	6ms
11.	GGSN-Destination Duplex BW	100Mbps
12.	GGSN-Destination Duplex Delay	8ms
13.	TCP increment UE1	Dn2
14.	TCP increment UE2	1
15.	UE – Bs distance	2 Km
16.	UE speed	0.1 m/s
17.	BS radius	2.5 Km

The experiment aimed to investigate the impact of varying the TCP increment coefficient of UE_1 on the throughput scaling in a bottleneck link. The value of Dn was set to increase from 0.1 to 1.0, and the throughput of UE1 was expected to scale with a ratio of $D_n : 1$ with respect to UE2. The experiment was conducted for 120 seconds, and the results were recorded in Table II.

Table II presents the first outcome of the experiment. The table displays the mean of throughput for each row, which was sampled every 5 seconds for 120 seconds. The experiment revealed that the increment coefficient (D_n) affected the throughput. Specifically, the ratio between increment coefficients of UE1 and UE2 approached the ratio between the throughput of UE1 and UE2 throughput, i.e., $Th1 : Th2 = Dn : 1$. For instance, when $D_n = 0.1$, the ratio was $Th_1 : Th_2 = 0.61384386 : 1$. This ratio mostly increased from $D_n = 0.2$ to 1, despite it was not a linear increase. The experiment showed that a higher increment coefficient (D_n) yields a higher throughput. The same behavior was observed for the other three users as well.

The results of throughput scaling with three UEs are presented in Table III and Figure 10. The network parameters are identical except for the addition of one more UE.

The results of the experiment conducted to observe the effect of increment coefficient (D_n) on throughput scaling with 3 UEs are presented in Table III and Figure 10. The table and figure represent the mean of throughput sampled every 5 s for 120 s. The experiment aimed to evaluate if the ratio between UE1 increment coefficient and UE2 or UE3 increment coefficient was consistent with the ratio between UE1 throughput and UE2 or UE3 throughput, i.e., $Th_1 : Th_2 = D_n : 1$ or $Th_1 : Th_3 = D_n : 1$. The ratio of $Th_1 : Th_2$ became 0.20877087:1 and $Th_1 : Th_3$ became 0.20277012:1 for $D_n = 0.1$. The results consistently increased for $D_n = \{0.2...1\}$. Despite the trend was not linear, it was observed that the higher increment coefficient had the higher throughput. The ratio between increment coefficient of UE1 and UE3 had a similar value to the ratio between UE1 throughput and UE3 throughput or simply $Th_1 : Th_3 = D_n : 1$. The throughput ratio between UE2 and UE3 was different due to the air interface environment's fluctuations, causing unstable bandwidth that affected throughput. In summary, increment coefficient (D_n) is able to regulate the throughput in the case of 3 UEs on a shared link.

B. *Weight Managing Evaluation*

This section is to assess the capability of a multipath host to adjust to changes in the environment.

Except for the experiments in part 4-B1 and 4-B2, UE1 was a host with mobile multipath connections, comprising of three HSDPA subflows. Under normal conditions, it consumed three times the throughput of a normal host. However, with eMPTCP, the throughput of UE1 was normalized to 1 to ensure fairness of the bottleneck. The advantage of having multiple connections was that the connection would be more stable since if a connection went down, other connections could back it up. The following figures show scenarios with both static and moving hosts. UE1 moved from the B1 area, which has a radius of 2.5 Km, towards B3 at a speed of 20m/s (72Km/h) for 200 s, covering a total distance of 4 Km and passing the border of the B1 and B2 areas. Table IV presents the parameters used in the experiment.

Several experiments were carried out to assess the impact of multipath and singlepath hosts on throughput. These experiments include: "All Static Singlepath Host on Bottleneck" in Section 4-B1, "Static Multipath Host on Bottleneck without throughput adjustment" in Section 4-B1, "Moving Singlepath Host on Bottleneck" in Section 4-B3, and "Moving Multipath Host on Bottleneck with and without the proposed eMPTCP" in Section 4-B4. The outcomes of these experiments are presented in the following figures.

1) *All Static Singlepath Host on Bottleneck*

The results of the simulation where a normal singlepath host (host 1) located in cell 1 competes with other normal single hosts (host 2, 3, 4) are depicted in Figure 12.

Figure 12 illustrates the results of the simulation where a normal singlepath host (host 1) in cell 1 competed with other normal single hosts (host 2, 3, and 4) for the bottleneck link. It is evident that all hosts competed fairly in the bottleneck link. The figure shows that the throughput of host 1 was similar to that of the other singlepath hosts for most of the running time. However, most of the connections experienced fluctuating throughput, which is normal since all hosts started with the same conditions. The purpose of this experiment was to demonstrate the actual conditions

TABLE II. Throughput Scaling Result

No	Dn	Throughput UE1(pps)	Throughput UE2(pps)	Th1:Th2
1	0.1	4463.67	7271.67	0.61384386
2	0.2	4680.33	7237	0.6467224
3	0.3	4966.33	7107	0.69879415
4	0.4	5122.33	6994.33	0.73235464
5	0.5	5469	6656.33	0.82162393
6	0.6	5980.33	6179.67	0.96774261
7	0.7	6275	5902.33	1.06313947
8	0.8	6431	5755	1.11746308
9	0.9	6483	5720.33	1.13332622
10	1	6535	5685.67	1.14938081

TABLE III. Throughput Scaling with 3 UE

no	Dn	Throughput UE1(pps)	Throughput UE2(pps)	Throughput UE2(pps)	Th1:Th2	Th1:Th3
1	0.1	1161.67	5564.33	5729	0.20877087	0.20277012
2	0.2	1326.33	5313	5815.67	0.24963862	0.22806143
3	0.3	1725	5044.33	5685.67	0.34196811	0.30339432
4	0.4	2201.67	4793	5460.33	0.45935114	0.40321189
5	0.5	2695.67	4550.33	5209	0.59241198	0.5175024
6	0.6	3181	4229.67	5044.33	0.75206813	0.63060902
7	0.7	3701	3943.67	4810.33	0.93846595	0.76938588
8	0.8	4238.33	3510.33	4706.33	1.20738791	0.90055946
9	0.9	4498.33	3415	4541.67	1.31722694	0.99045725
10	1	4619.67	3588.33	4247	1.28741504	1.087749

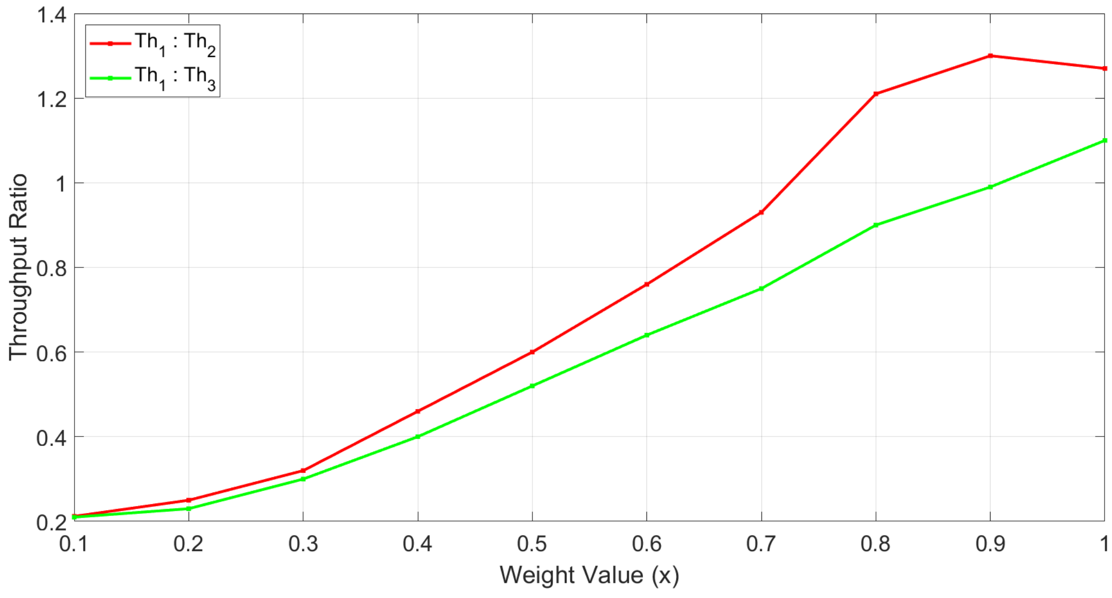

Figure 10. Throughput Scaling Result with 3 UEs

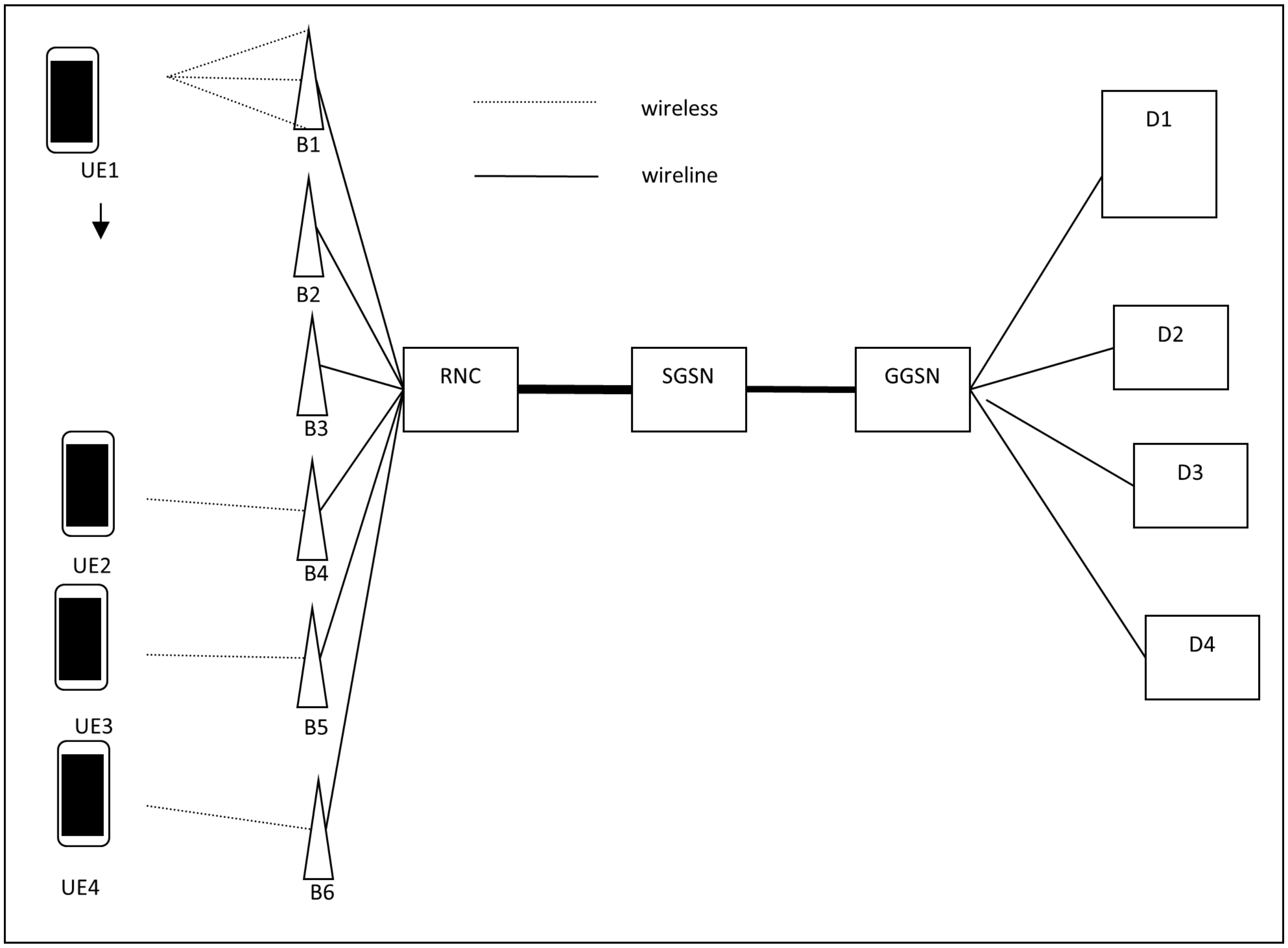

Figure 11. Multipath Host Adaptability Experiment Scenario

TABLE IV. Network Parameters for Weight Managing Evaluation

no	Parameters	Value
1.	BS Downlink BW	32kbps
2.	BS Downlink TTI	10ms
3.	BS Uplink BW	32kbps
4.	BS Uplink TTI	10ms
5.	BS-RNC(Iub) Uplink/Downlink BW	155Mbps
6.	BS-RNC(Iub) Uplink/Downlink Delay	15ms
7.	RNC-SGSN Duplex BW(Bottleneck)	0.23Mbps
8.	RNC-SGSN Duplex Delay	6ms
9.	GGSN-SGSN Duplex BW(Bottleneck)	0.23Mbps
10.	GGSN-SGSN Duplex Delay	6ms
11.	GGSN-Destination Duplex BW	100Mbps
12.	GGSN-Destination Duplex Delay	8ms
13.	TCP increment UE1	D12, D22, D32
14.	TCP increment UE2,3,4	1
15.	UE – Bs distance	2 Km
16.	UE1 speed	0 and 20 m/s
17.	UE2,3,4 speed	0.1 m/s
18.	BS radius	2.5 Km

of the bottleneck used by the same singlepath TCP. At certain periods, all singlepath TCPs experience failures due to random lossy mobile. In the worst-case scenario, the throughput drops below 1000 packets per second. This condition will be worse for moving singlepath hosts, as evaluated in experiment 4-B3. In the following scenario, eMPTCP is employed to overcome this phenomenon.

2) Static Multipath on Bottleneck Using MPTCP without Adjustment

The main objective of this study was to find a method to achieve fair sharing of throughput among multiple hosts. The primary challenge that needed to be addressed is demonstrated in Figure 13, which shows the result of a multipath host competing with a normal singlepath host in the absence of the throughput adjustment.

Figure 13 indicates that the main problem of the study was the domination of multipath host (host1) over the bandwidth share when competing with normal singlepath host without using the throughput adjustment. The multipath host consumed almost three times the throughput of a normal singlepath host, and the latter only consumed 4000-6000 packets/second of throughput, which was significantly lower than the multipath host. Each Multipath TCP subflow grew at the same rate as other singlepath TCP, resulting in a total throughput of Multipath TCP that was also three times that of normal singlepath TCP. However, the singlepath throughput was lower than the previous singlepath throughput (Fig. 12) since Multipath TCP was over-aggressive.

3) Moving Singlepath

The simulation involved a normal singlepath host moving at a speed of 20m/s from area BS1 to other BSs. As shown in Figure 14, the moving host experienced a smaller and fluctuating throughput value due to the undeterministic nature of mobile signals. The fluctuations were caused by varying signal strength, which is typical of mobile connections. The handover process, which occurred around 130-160 s, resulted in a significant drop in throughput.

The above figure demonstrates that during the handover condition, singlepath TCP experiences a significant loss in throughput. The throughput drops to almost zero. During handover, a physical connection (mobile air interface) must be terminated and a new one must be established, causing a drop in data communication, which affects the throughput. With Multipath TCP, however, each subflow uses a different air interface, so the handover has a different effect. The subsequent experiments demonstrate this effect.

4) Moving Multipath

The following figures illustrate the simulation results of a moving multipath host and some normal singlepath hosts during the experiments to test the effectiveness of the proposed eMPTCP in handling fluctuating bandwidth in a mobile network. Similar to the previous figure, handover events occur at certain points during the simulation. These handovers have different effects on the moving multipath host and the static singlepath hosts on the bottleneck link.

Based on Figure 15, it can be concluded that the multipath host dominates the throughput and bandwidth in almost all running time. During the entire simulation running time, the throughput was much higher than that of the normal single host. The throughput value only dropped in the handover period, which was around 130-150 s. Since the subflows are not weighted, the unweighted Multipath TCP has three subflows that consume three times the bandwidth share of the bottleneck.

According to Figure 15, at the beginning of the simulation, the unweighted multipath consumes high bandwidth, causing the singlepath throughput to become very low. In a certain period, 140-160 s, multipath suffers great loss caused by handover. The other singlepath flows take the chance to consume more bandwidth, which causes the singlepath to have a higher throughput. But the singlepath flows only take a little time period to grow, since the remaining period of the simulation running time, the un-weighted Multipath dominates the bandwidth consumption on the bottleneck. Overall, it is clear that the multipath host violated the fairness.

In this experiment (Figure 16), a bandwidth limit of 50000 was implemented. The extremely high throughput of the multipath host in the beginning, which was out of our analysis since it was still in the slowstart phase, made the window size become very high. The result was better than the previous experiment in terms of fairness. The figure shows that the multipath host had lower throughput than the single path hosts because the multipath host was moving, and the effect of channel conditions influenced the loss rate, causing the throughput to become lower. Especially in period 125-150 s, the throughput dropped significantly since host 1 had to be handed-over from area BS1 to BS2. However, the figure shows that the throughput of the other singlepath hosts was higher than in the previous experiment since the multipath host was less dominating the bandwidth.

The eMPTCP suffered more loss than the static singlepath since the multipath was moving, which caused fading on its air interface. Regarding the throughput of the unweighted multipath in the previous experiment, the weighted multipath successfully reduces the domination so that the singlepath TCP flows obtain more throughput than in the previous simulation. On the other hand, the eMPTCP does not suffer higher loss than singlepath TCP (Figure 14) on the handover condition since the Multipath TCP has more redundant physical air interfaces than singlepath TCP.

The results from this experiment (Figure 17) showed some slight differences compared to the previous one. In this case, a bwMax of 110000 was implemented, which was the highest bwe value of a single path HSDPA air interface. The extremely-high throughput of the multipath host in the beginning was also not analyzed since it was still in the

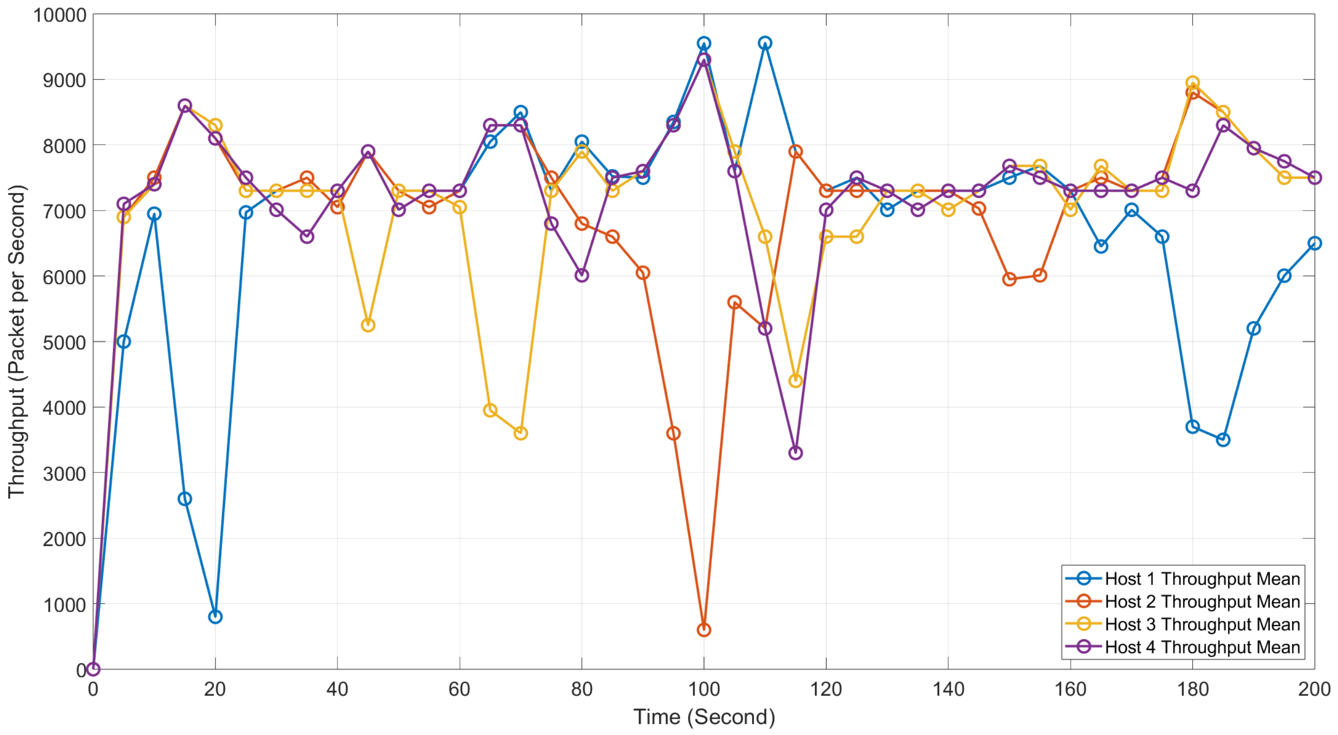

Figure 12. All Static Singlepath Hosts Throughput Comparison

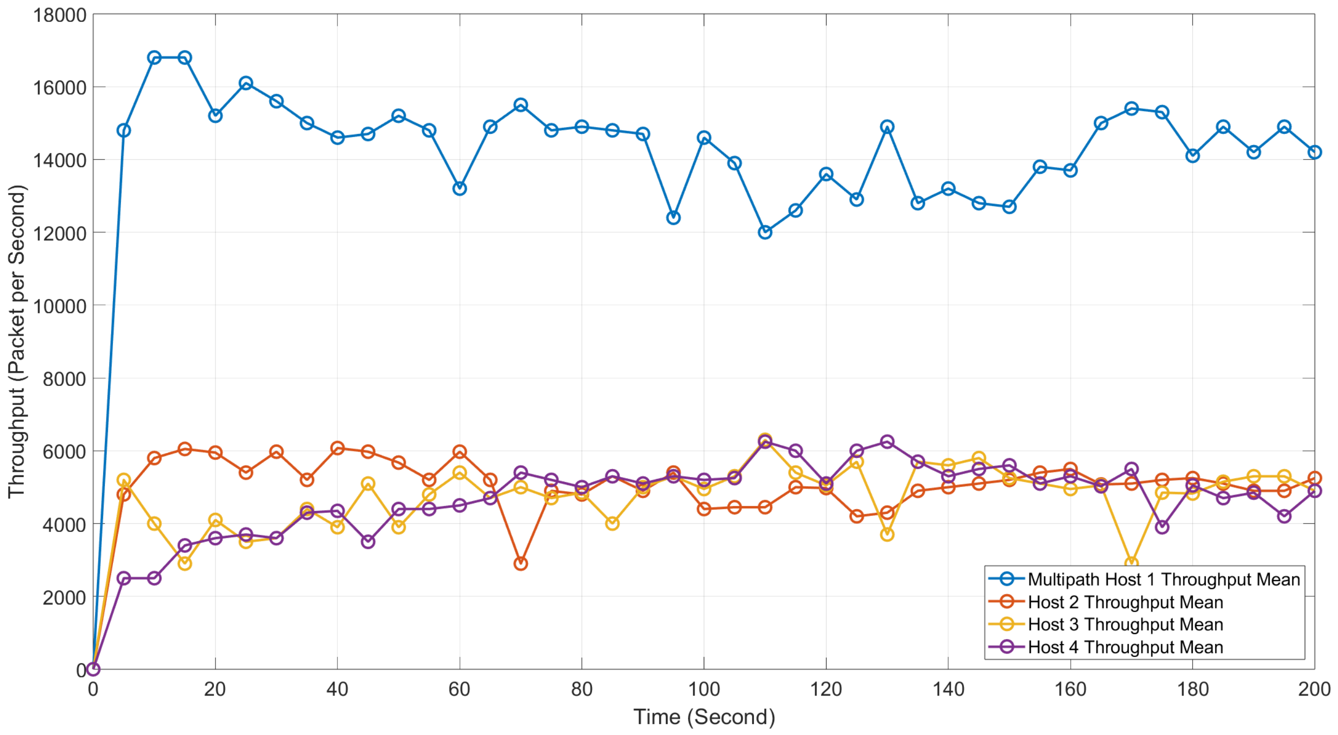

Figure 13. Static Multipath and Singlepath Hosts Throughput Comparison without the proposed eMPTCP

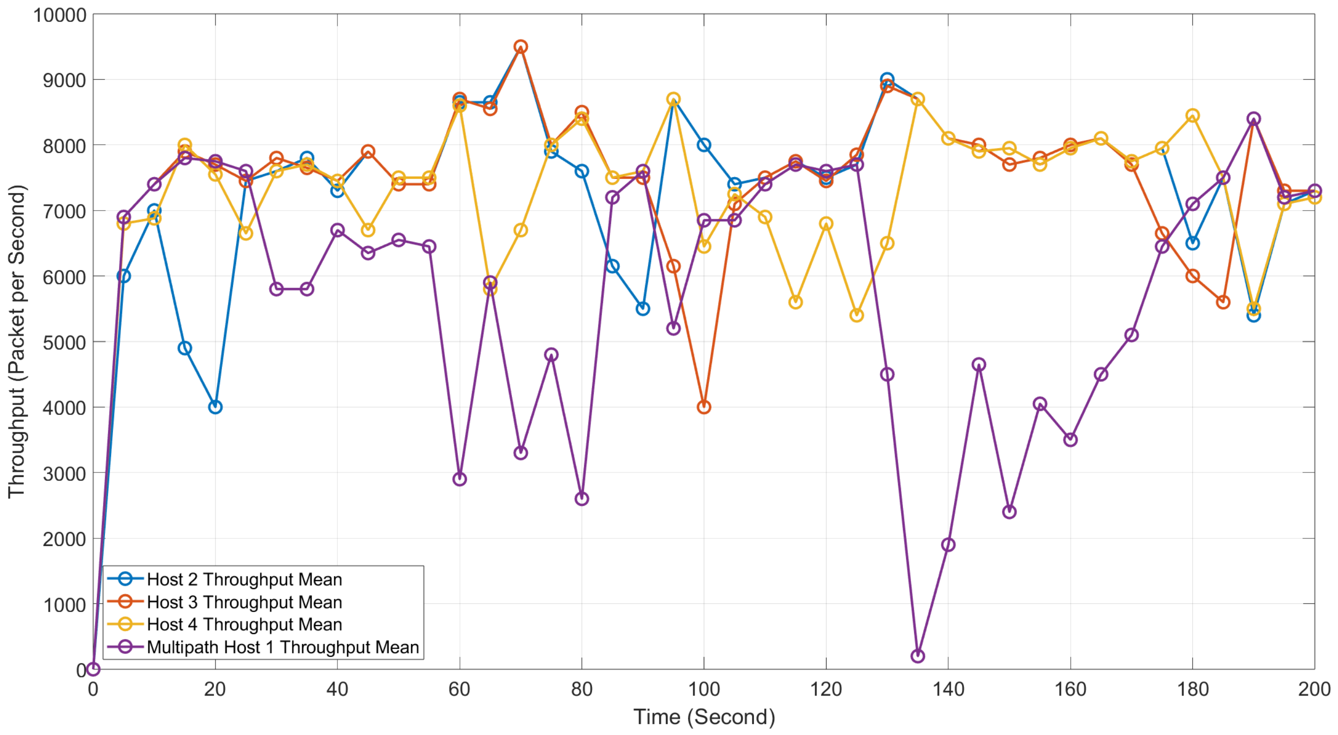

Figure 14. Moving Singlepath and Static Singlepath Hosts Throughput Comparison

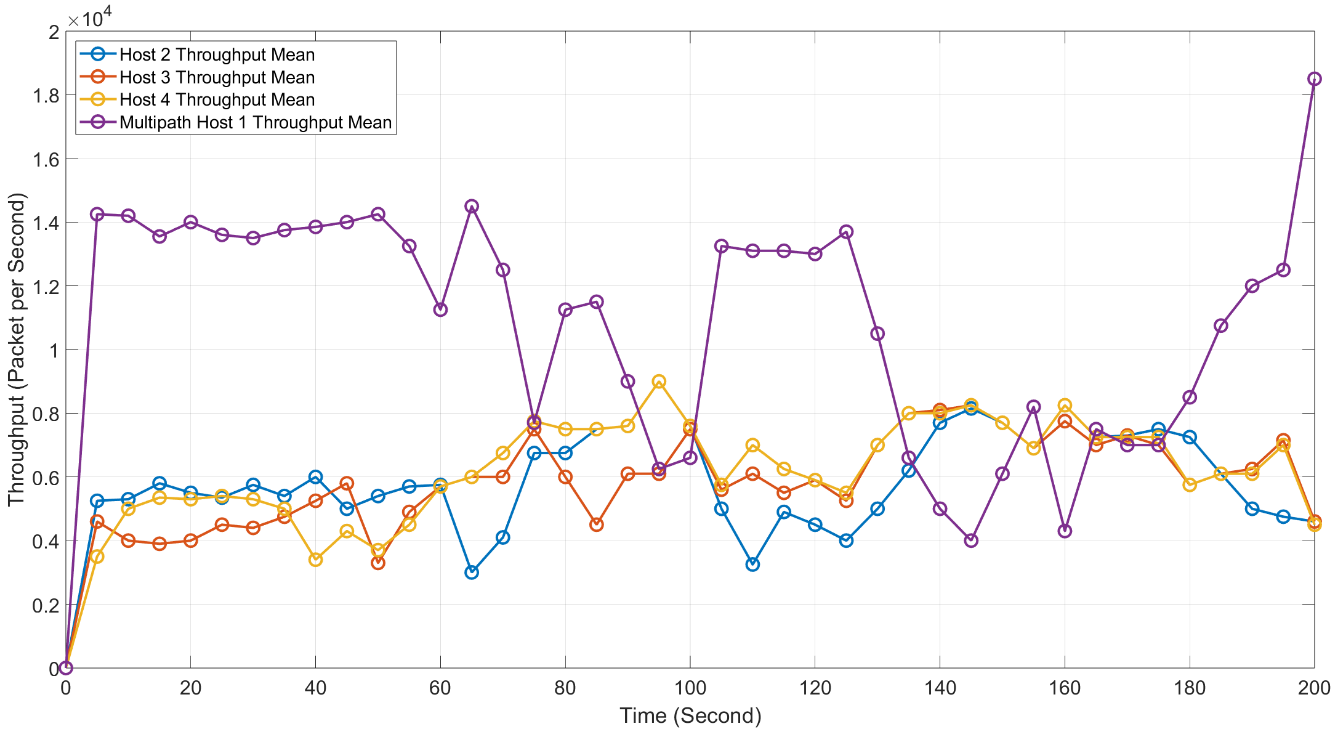

Figure 15. Moving Multipath and Static Singlepath Hosts Throughput Comparison without the proposed eMPTCP

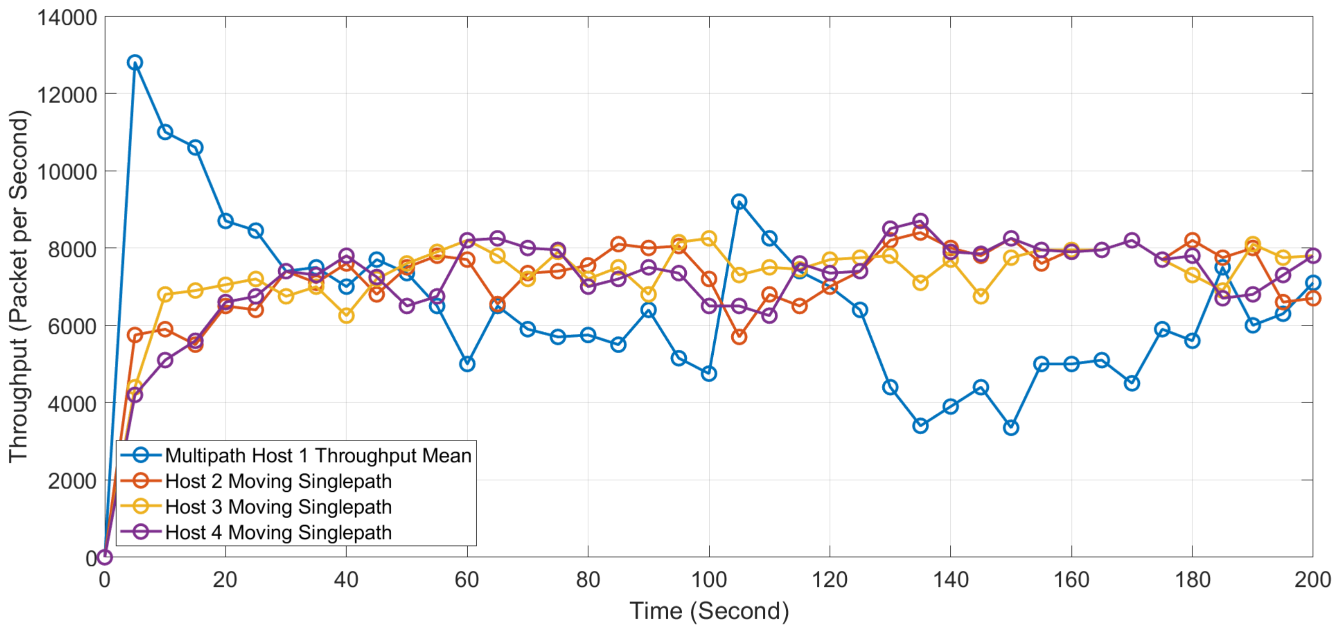

Figure 16. Moving Multipath and Static Singlepath Hosts Throughput Mean Comparison with the proposed eMPTCP (bwmax=50000)

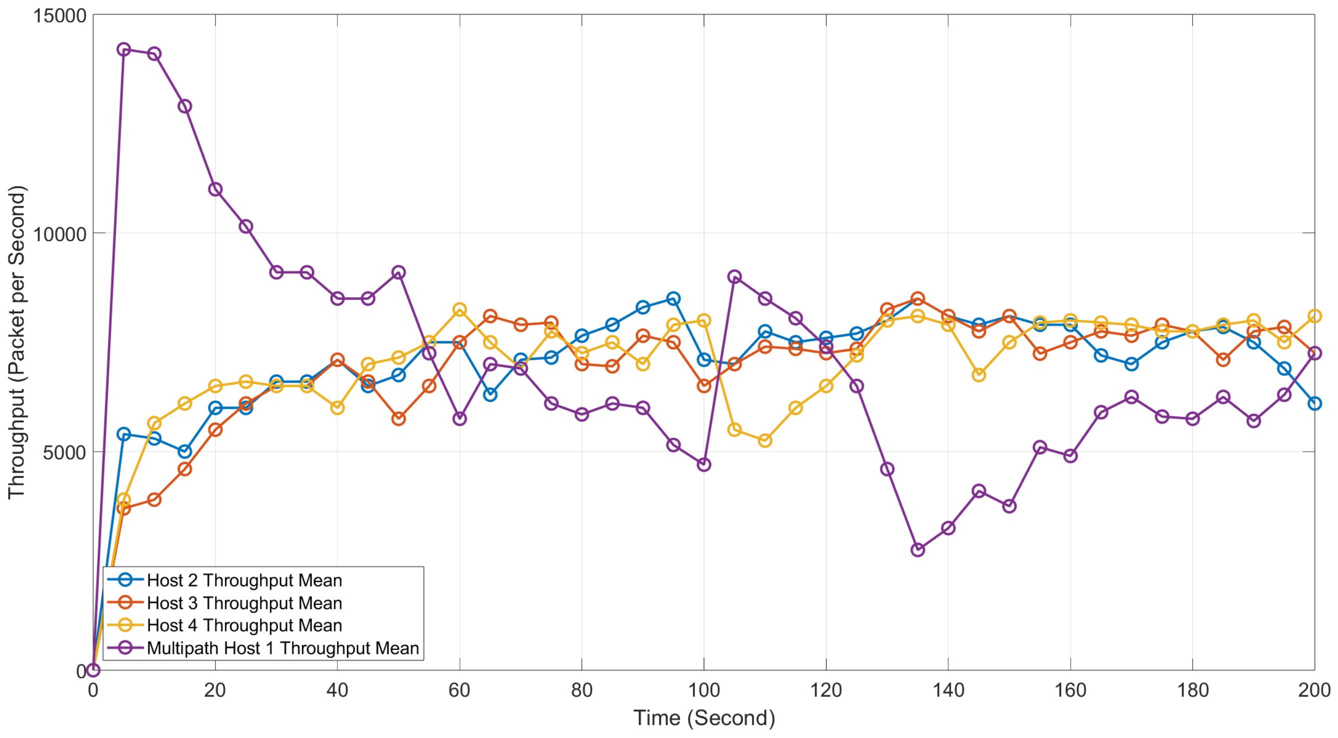

Figure 17. Moving Multipath and Static Singlepath Hosts Throughput Mean Comparison with the proposed eMPTCP (bwmax=110000)

slowstart phase, which made the window size very high. However, in the middle of the simulation, the multipath throughput dropped due to the host moving and the effect of channel conditions causing a higher error rate, resulting in a lower throughput. Similarly to the previous figure, the throughput dropped significantly in period 125-150 s due to the handover of host 1 from area BS1 to BS2.

The main difference from the previous experiment was that the throughput value of the multipath host was slightly higher than before. This was due to the throughput adjustment used, where higher bwMax leads to a higher weight value, which in turn leads to a higher throughput. This method influences the total throughput, as shown in Figures 16 and 17.

5. Conclusions

Smarphones currently account for a significant proportion of internet traffic. Since most of the devices are multihomed mobile devices with multiple NIs, MPTCP can be used to improve the connection resilience without damaging the fairness. This paper proposed the eMPTCP to maintain the connection fairness and achieve resilience during the device mobility. The proposed method consists of throughput adjustment and weight managing methods. The proposed method is further simulated under cellular networks both static and mobile condition. Inspite of network distortion due to the air interface environment's fluctuations, the throughput adjustment using increment coefficient is able to govern mobile device connection throughput. Further, during the network mobility assessment, the eMPTCP successfully preserves the fairness so that the singlepath TCP flows maintain the regular throughput. In addition to that, the eMPTCP is able to improve the mobile device resilience compared to singlepath TCP during the handover condition since the Multipath TCP has more redundant physical air interfaces than singlepath TCP. The main limitation of this method is the availability of multiple network interfaces and the communication among interfaces will negatively increase the overhead computation in the kernel.

6. Acknowledgement

This research is supported by PPM of Telkom University under grant No. 924/PNLT3/PPM/2022 and No. 926/PNLT3/PPM/2022.

References

[1] S. Szecsei, "What percentage of internet traffic is mobile in 2023," Mar 2023. [Online]. Available: https://capitalcounselor.com/what-percentage-of-internet-traffic-is-mobile/

[2] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin, "A first look at traffic on smartphones," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, 2010, pp. 281–287.

[3] E. Press, "Mobile data traffic surpasses voice," 2010.

[4] N. Wood, "Mobile data traffic growth 10 times faster than fixed over next five years," *Total Telecom: http://www. totaltele. com/view. aspx*, 2009.

[5] M. Zaharia and S. Keshav, "Fast and optimal scheduling over multiple network interfaces," *University of Waterloo Technical Report, CS-2007-36*, 2007.

[6] C. Paasch, S. Barre *et al.*, "Multipath tcp implementation in the linux kernel," 2014.

[7] L. Chao, C. Wu, T. Yoshinaga, W. Bao, and Y. Ji, "A brief review of multipath tcp for vehicular networks," *Sensors*, vol. 21, no. 8, p. 2793, 2021.

[8] W. Li, H. Zhang, S. Gao, C. Xue, X. Wang, and S. Lu, "Smartcc: A reinforcement learning approach for multipath tcp congestion control in heterogeneous networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2621–2633, 2019.

[9] R. K. Chaturvedi and S. Chand, "An adaptive and efficient packet scheduler for multipath tcp," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 45, pp. 349–365, 2021.

[10] H. H. Nuha, F. A. Yulianto *et al.*, "Wireless multi-path tcp westwood+ modification to achieve fairness in hsdpa," in *2010 Fourth UKSim European Symposium on Computer Modeling and Simulation*. IEEE, 2010, pp. 402–407.

[11] Y. Deguchi, A. Kobayashi, Y. Tarutani, Y. Fukushima, and M. Nakamura, "Throughput fairness in congestion control of multipath TCP," in *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2022, p. 1–4.

[12] G. Hasegawa and M. Murata, "Survey on fairness issues in TCP congestion control mechanisms," *IEICE transactions on communications*, vol. 87, no. 5, p. 1358–1366, 2004.

[13] S. Liu, W. Lei, W. Zhang, and X. Song, "Mpmtp-ar: Multipath message transport protocol based on application-level relay," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 11, no. 3, pp. 1406–1424, 2017.

[14] J. Huang, W. Li, Q. Li, T. Zhang, P. Dong, and J. Wang, "Tuning high flow concurrency for mptcp in data center networks," *Journal of Cloud Computing*, vol. 9, pp. 1–15, 2020.

[15] Q. Peng, A. Walid, J. Hwang, and S. H. Low, "Multipath tcp: Analysis, design, and implementation," *IEEE/ACM Transactions on networking*, vol. 24, no. 1, pp. 596–609, 2014.

[16] Z. Fang, J. Wang, Y. Ren, Z. Han, H. V. Poor, and L. Hanzo, "Age of information in energy harvesting aided massive multiple access networks," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 5, pp. 1441–1456, 2022.

[17] X. Hou, Z. Ren, J. Wang, W. Cheng, Y. Ren, K.-C. Chen, and H. Zhang, "Reliable computation offloading for edge-computing-enabled software-defined iov," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7097–7111, 2020.

[18] M. Honda, "Bidimensional-probe multipath congestion control for shared bottleneck fairness," Ph.D. dissertation, Citeseer, 2009.

[19] M. Gerla, M. Y. Sanadidi, R. Wang, A. Zanella, C. Casetti, and S. Mascolo, "Tcp westwood: Congestion window control using bandwidth estimation," in *GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No. 01CH37270)*, vol. 3. IEEE, 2001, pp. 1698–1702.

[20] W. Stevens, “Tcp slow start, congestion avoidance, fast retransmit, and fast recovery algorithms,” Tech. Rep., 1997.

[21] J. Postel, “Rfc0793: Transmission control protocol,” 1981.

[22] M. Welzl, *Network congestion control: managing internet traffic*. John Wiley & Sons, 2005.

[23] M. Scharf and A. Ford, “Multipath tcp (mptcp) application interface considerations,” Tech. Rep., 2013.

[24] A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar, “Architectural guidelines for multipath tcp development,” Tech. Rep., 2011.

[25] C. Raiciu, M. Handley, and D. Wischik, “Coupled congestion control for multipath transport protocols,” Tech. Rep., 2011.

Hilal H. Nuha is an associate professor at School of Computing, Telkom University, Bandung, Indonesia. He has published many papers on topics such as computer communication, data processing, seismic data compression, and wind speed estimation.

Fazmah Arif Yulianto is a lecturer and researcher at Telkom University in Indonesia. He has a doctoral degree from Institut Teknologi Bandung (ITB). His research interests are networking, multimedia, security, and real-time computing.

Hendrawan is a lecturer and researcher at Institut Teknologi Bandung (ITB) in Indonesia. He has a doctoral degree from University of Essex in United Kingdom1. He is an expert in telecommunication, wireless network, digital TV broadcast, and video compression. He has published many papers and received several awards for his work.