MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

# Modeling Multipath TCP Over Heterogeneous WiFi and 5G Networks

Guo, Jianlin; Parsons, Kieran; Nagai, Yukimasa; Sumi, Takenori; Sakaguchi, Naotaka; Wang, Pu; Orlik, Philip V.

TR2025-084 June 10, 2025

**Abstract**

As the number of wireless devices supporting multiple communication interfaces increases, the connection redundancy is being considered for efficient bandwidth utilization and QoS improvement. Accordingly, network technologies must adapt to emerging multi-interface devices to improve network performance. Multipath TCP (MPTCP) is default multipath transport protocol desired for networks with multi-interface de- vices and has achieved success in computer networks. However, it has not been well studied for wireless networks, especially for carrier sense multiple access (CSMA) based wireless networks, which present great challenges to round trip time (RTT) computation and multipath scheduling. This paper introduces MPTCP techniques for heterogeneous WiFi and 5G networks. We first model a proposed 5-state congestion control algorithm and WiFi CSMA function. We then present an innovative RTT computation method and a novel loss-ware multipath scheduling mechanism. We evaluated the proposed MPTCP techniques under varying network configurations. Our MPTCP can significantly outperform conventional MPTCP.

*IEEE International Conference on Communications (ICC) 2025*



Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

# Modeling Multipath TCP Over Heterogeneous WiFi and 5G Networks

Jianlin Guo*, Kieran Parsons*, Yukimasa Nagai†, Takenori Sumi†, Naotaka Sakaguchi†, Pu Wang*
and Philip Orlik*
*Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139, USA
†Information Technology R&D Center, Mitsubishi Electric Corporation, Kamakura, Kanagawa 2478501, Japan

***Abstract*—As the number of wireless devices supporting multiple communication interfaces increases, the connection redundancy is being considered for efficient bandwidth utilization and QoS improvement. Accordingly, network technologies must adapt to emerging multi-interface devices to improve network performance. Multipath TCP (MPTCP) is default multipath transport protocol desired for networks with multi-interface devices and has achieved success in computer networks. However, it has not been well studied for wireless networks, especially for carrier sense multiple access (CSMA) based wireless networks, which present great challenges to round trip time (RTT) computation and multipath scheduling. This paper introduces MPTCP techniques for heterogeneous WiFi and 5G networks. We first model a proposed 5-state congestion control algorithm and WiFi CSMA function. We then present an innovative RTT computation method and a novel loss-ware multipath scheduling mechanism. We evaluated the proposed MPTCP techniques under varying network configurations. Our MPTCP can significantly outperform conventional MPTCP.**

***Index Terms*—Multipath TCP, WiFi CSMA, RTT computation, congestion control, heterogeneous wireless networks.**

## I. INTRODUCTION

Wireless devices with multiple communication interfaces are referred to as multi-interface (MI) devices. Smartphones and laptop computers are typical multi-interface devices. Recently, wireless IoT devices such as smart meters can also support multiple communication interfaces. Therefore, the connection redundancy provides an opportunity to improve network performance, especially for multi-hop wireless networks. Furthermore, the nodes in CSMA based wireless networks typically form mesh topology, where multiple communication paths can be established between nodes. These paths can be simultaneously used for communications.

MPTCP specified in IETF RFC 8684 is the default multipath transport protocol to allow simultaneous use of multiple communication interfaces, which can achieve higher throughput and complete transmissions in a shorter time [1]. However, despite the success of MPTCP in computer networks, its deployment over wireless networks is not well studied, especially over CSMA based wireless networks, in which random backoff delay incurs significant challenges for RTT computation and path scheduling. The key issue for MPTCP is the head of line (HoL) blocking caused by out-of-order packet arriving due to path heterogeneity such as RTT, path bandwidth and path loss.

This paper studies MPTCP over multi-hop heterogeneous WiFi and 5G networks. We propose a 5-state congestion control algorithm, an innovative RTT computation method, and a novel loss-aware MPTCP path scheduling mechanism. We evaluated the proposed MPTCP techniques under varying network configurations and observed interesting insights. To the best of our knowledge, we are the first to model MPTCP over heterogeneous WiFi and 5G networks.

## II. RELATED WORKS

Congestion control is critical for MPTCP. Although there are alternative congestion control methods such as the Opportunistic Linked Increases Algorithm (OLIA), the standard NewReno algorithm specified in IETF RFC 6582 is a default congestion controller for MPTCP.

Path scheduling is the key for MPTCP. The minimal RTT (MinRTT) is the default MPTCP scheduler. However, it faces HoL blocking issue since it is barely based on RTT. There are scheduling methods proposed to enhance the MinRTT scheduler by considering other path metrics. Authors in [2] propose a delay-aware packet scheduling (DAPS), which aims to reduce the receiver's buffer blocking time. Work [3] presents a blocking estimation-based MPTCP scheduler (BLEST) to minimize HoL blocking. Paper [4] proposes a loss-aware throughput estimation scheduler and a method to compute the number of packets that can be transmitted over a path in a scheduling round. However, these works do not address the RTT computation, which is required by MPTCP path scheduling and challenging to compute in wireless networks, especially in CSMA based wireless networks. In addition, these works use pre-configured small network topology for performance evaluation without considering network dynamics. Accordingly, the field of MPTCP scheduling must be revisited [5]. Authors of this paper have proposed a method to compute RTT over heterogeneous IEEE 802.15.4 and 5G networks [6]. However, WiFi CSMA, i.e., IEEE 802.11 Distributed Coordination Function (DCF), is more complicated than IEEE 802.15.4 CSMA, e.g., IEEE 802.15.4 CSMA does not sense channel in backoff periods and does not suspend backoff process, and WiFi CSMA, on the other hand, senses channel in each backoff time slot and suspends backoff process if the channel is busy. Accordingly, MPTCP over WiFi based wireless networks needs to be studied.

There are works that model WiFi CSMA as Markov chain. The pioneer work is Bianchi model [7], which models WiFi backoff process but does not consider backoff suspension and immediate channel access. Paper [8] extends Bianchi's work to model WiFi CSMA with backoff suspension but does not include immediate channel access. Recent work [9] models WiFi CSMA with immediate channel access but does not incorporate backoff suspension.

This paper proposes MPTCP techniques over heterogeneous wireless networks consisting of a data center (DC), WiFi data nodes and 5G data nodes that support both 5G and WiFi communication interfaces, i.e., MI nodes. Considering that the device-to-device (D2D) communication in 5G is not yet fully supported, we assume 5G nodes can communicate with WiFi nodes and DC via WiFi interface and 5G base station, respectively. DC is considered as a MI node.

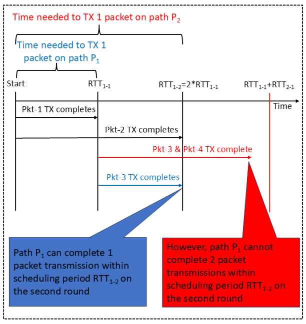


Fig. 1: Necessity of Maintaining $cwnd$ Without Packet Loss

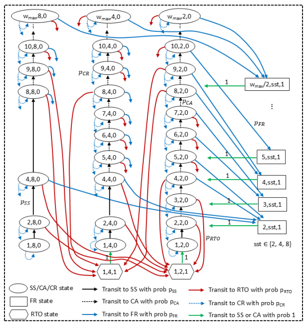

Fig. 2: Markov Chain Model of Our 5-State NewReno Algorithm

## III. 5-State NewReno Congestion Control Algorithm and Its Markov Chain Model

Standard NewReno is a 4-state algorithm and uses congestion window ($cwnd$), slow start threshold ($sst$) and receiver window ($rwnd$) to control traffic congestion. The $cwnd$, denoted as $w$ for short, limits the number of packets can be transmitted over a path in a scheduling round and the $rwnd$ indicates amount of data receiver is willing to accept. NewReno algorithm starts in slow start (SS) state with $w$ set to $w_{min}$ and $sst_{start}$ set to the largest advertised $rwnd$ or a value based on network path. If there is no packet loss in a scheduling round, $w$ is doubled in next scheduling round. When $w$ reaches $sst$, the algorithm transits to congestion avoidance (CA) state, in which $w$ is incremented by 1 in each scheduling round until $w$ reaches $w_{max}$. In either SS state or CA state, if packet loss occurs in a scheduling round, the algorithm transits to fast retransmit (FR) state if the loss is triggered by three duplicate ACKs and the lost packets can be recovered within the remaining window $w$ or otherwise to retransmit timeout (RTO) state. If algorithm transits to FR state, both $sst$ and $w$ are set to $w/2$. If algorithm transits to RTO state, $sst$ is set to $w/2$ and $w$ is set to $w_{min}$. The state transition probabilities are given as: $p_{SS} = p_{CA} = p(0|w)$, $p_{RTO} = \begin{cases} \sum_{i=1}^{w-3} p(i|w)(1-(1-l)^i) + \sum_{i=w-2}^{w} p(i|w), & w \geq 4 \\ 1 - p(0|w), & w < 4 \end{cases}$

$$p_{FR} = \begin{cases} 1 - p_{RTO} - p(0|w), & w \geq 4 \\ 0, & w < 4 \end{cases}$$

where $l$ is the packet loss probability and $p(x|w) = \binom{w}{x} l^x (1-l)^{w-x}$ is the probability of $x$ packet loss in $w$ packets [10], which models the $w$ change behavior of TCP connections.

NewReno algorithm was designed for computer networks with relatively stable network environment. To adapt to dynamic wireless networks, authors of this paper proposed an adaptive NewReno (A-NewReno) algorithm [6], which provides three adaptations: (1) $w_{min}$ and $w_{max}$ adaptation, (2) RTO timer adaptation and (3) $w$ update frequency adaptation. This paper further enhances NewReno algorithm by proposing a new state. Specifically, in NewReno algorithm, $w$ increases unless packet loss occurs, which is inefficient. We first illustrate the necessity of maintaining or reducing $w$ even without packet loss and then introduce the new stage.

A TCP packet is considered as data frame at WiFi MAC layer, indicating a WiFi receiver will transmit a MAC ACK frame at Short Interframe Space (SIFS) time before forwarding the received TCP packet up to TCP layer. Recall that WiFI CSMA enables an immediate channel access mechanism in which if channel is idle for more than DCF Interframe Space (DIFS) time, a WiFi device can transmit frame immediately without random backoff. Denote as $\mathrm{RTT}_{x-h}$ the RTT to transmit $x$ packets over a $h$-hop WiFi path and $\mathrm{T}_P$ the time to deliver packet $P$. Assuming WiFi channel is idle and delayed TCP ACK mechanism is applied. Ignoring packet propagation time and TCP/IP layer packet processing time, we have $\mathrm{RTT}_{1-1} = 2\mathrm{DIFS} + 2\mathrm{SIFS} + 2\mathrm{T}_{MAC\ ACK} + \mathrm{T}_{TCP\ Data} + \mathrm{T}_{TCP\ ACK}$, $\mathrm{RTT}_{1-2} = 2\mathrm{RTT}_{1-1}$, and $\mathrm{RTT}_{2-1} = \mathrm{RTT}_{1-1} + \mathrm{DIFS} + \mathrm{T}_{TCP\ Data} + \mathrm{SIFS} + \mathrm{T}_{MAC\ ACK}$.

Fig. 1 illustrates the necessity of maintaining $w$ under no packet loss condition, where path $\mathrm{P}_1$ is a 1-hop path and path $\mathrm{P}_2$ is a 2-hop path, these two paths do not interfere with each other. For MPTCP scheduling, RTT of the slower path is typically used as scheduling period. Accordingly, $\mathrm{RTT}_{1-2}$ of path $\mathrm{P}_2$ is scheduling period. Assume $w_1 = w_2 = 1$ initially, in the first scheduling round, packet-1 is sent on path $\mathrm{P}_1$ and packet-2 is sent on path $\mathrm{P}_2$. Path $\mathrm{P}_1$ completes its first round at time $\mathrm{RTT}_{1-1}$, $w_1$ is then increased to 2 for the second round. However, path $\mathrm{P}_1$ can not deliver 2 packets within remaining $\mathrm{RTT}_{1-1}$ time. As a result, path $\mathrm{P}_1$ is idle for $\mathrm{RTT}_{1-1}$ time, which is a waste. Had $w_1$ remained to 1, path $\mathrm{P}_1$ can deliver packet-3 in remaining $\mathrm{RTT}_{1-1}$ time. Similarly, we can show the necessity of decreasing $w_1$.

To address this need, we propose a new state named as $cwnd$ relaxation (CR) for no packet loss case, in which $sst$ remains, and $w$ remains or decreases. Consider a path P, state transition probability $p_{CR}$ depends on scheduling period duration $\mathrm{D}_p$ of the path P, scheduling round $r$ within the scheduling period, congestion window $w(r)$ on the r-th round and times spent $\mathrm{T}_p(i)$ ($i$=1,2,$\cdots$,r) by path P up to the r-th scheduling round. The probability that r+1 round transits to CR state is $p_{CR} = p(0|w(r))p(T_p(1) + T_p(2) + \cdots + T_p(r+1) > \mathrm{D}_p | T_p(1) + T_p(2) + \cdots + T_p(r) < \mathrm{D}_p)$.

Fig. 2 illustrates Markov chain model of our proposed 5-State NewReno algorithm state transition with corresponding probability, where each state is represented by three state variables $w$, sst, and $l$. The sst values 2, 4, and 8 are used for illustration. Solid black lines indicate SS state transition with probability $p_{SS}$, $w = 2w$, sst = sst and $l = l = 0$. Dash black lines indicate CA state transition with probability $p_{CA}$, $w = w{+}1$, sst = sst and $l = l = 0$. Solid green lines indicate state transitions from FR or RTO states to SS or CA states with probability 1, $w = w$, sst = sst and $l$ transiting from 1 to 0. Dash blue lines indicate state transitions from SS or CA states to CR states with probability $p_{CR}$, sst = sst, $l = l = 0$ and $w \leq w$, where " $=$ " sign indicates staying at current state with $w = w$ and " $<$ " sign indicates transiting to a lower state with $w < w$. Solid long and short blue lines indicate state transitions to FR states with probability $p_{FR}$, $l$ transiting from 0 to 1, sst = $w/2$ and $w = w/2$. Solid long and short red lines indicate state transitions to RTO states with probability $p_{RTO}$, $l$ transiting from 0 to 1, sst = $w/2$ and $w = w_{min}$, where depending on the value of sst, the state transitions can move to left column, stay in current column, or move to right column.

The proposed Markov chain model is used to compute the expected congestion window size $w$, which is then used in MPTCP scheduling method.

## IV. Markov Chain Model of WiFi CSMA Function

WiFi supports CSMA, i.e., DCF, by default, which is contention based channel access mechanism and mandatory for WiFi devices. WiFi CSMA specifies that a device can make immediate transmission if the channel is idle for more than DIFS time. Otherwise, the device has to perform random backoff process, which can take place only after channel is free for DIFS time. Depending on the length of a frame, WiFi transmitter can backoff up to dot11ShortRetryLimit or aLongRetryLimit times, denoted as $r$ for short. Upon receiving a frame that requires ACK, WiFi receiver transmits an ACK frame after SIFS time. Backoff counter, i.e., the number of backoff time slots, is uniformly drawn in the range $[0, W_i]$, where $W_i$ is the contention window (CW) size at the backoff stage $i$ with $CW_{min} \leq W_i \leq CW_{max}$ and $0 \leq i \leq r$, where $W_i = 2^i W, i \in [0, m]$ and $W_i = 2^m W$, if $r > m$ and $i \in [m+1, r]$ with $W = CW_{min}$ being the minimum CW size and $m$ being determined using the maximum CW size as $CW_{max} = 2^m W$. WiFi CSMA performs channel sense in each backoff time slot. Backoff counter is decreased by 1 for each idle time slot and suspended on the busy channel.

WiFi CSMA random backoff delay can be significant due to backoff suspension and therefore, must be considered in RTT computation. To that end, we need a fully functional WiFi CSMA model. However, the existing models found either consider retry limit $r \leq m$ only or lack backoff suspension or miss immediate channel access mechanism that can greatly impact WiFi channel access delay in RTT computation under the non-saturated traffic condition, the typical traffic scenario in practical wireless networks.

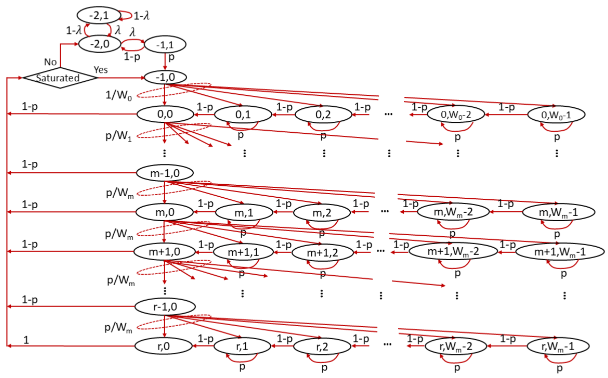

Fig. 3: Proposed Markov Chain Model for Fully Functional WiFi CSMA

This paper presents a Markov chain model for WiFi CSMA by incorporating immediate channel access, flexible $r$, backoff suspension, saturated and non-saturated traffic scenarios. Fig. 3 shows the proposed model, where each state is represented by two random variables, backoff stage $s(t)$ and backoff counter $b(t)$. Let $b_{i,k} = \lim_{t\to\infty} Pr(s(t) = i, b(t) = k)$, where $i \in [0, r], k \in [0, W_i]$. To model immediate channel access, we introduce two new states $b_{-1,0}$, the backoff start state due to the busy channel or new packet transmission, and $b_{-1,1}$, the state to attempt immediate channel access. To model non-saturated condition, work [8] introduced another two states $b_{-2,0}$, the data available state with probability $\lambda$, and $b_{-2,1}$, the data not available state with probability $1 - \lambda$. All $b_{i,k}$ can be expressed using $b_{0,0}$. Adopting the assumption in [7] by assuming the collision probability $p$ of a transmitted packet is constant and independent of the number of retransmissions this packet has suffered in the past, we have following state transition probabilities:

$$Pr(i, k|i, k+1) = 1 - p, k \in [0, W_i - 2], i \in [0, r]$$
$$Pr(i, k|i, k) = p, k \in [0, W_i - 1], i \in [0, r]$$
$$Pr(i, k|i-1, 0) = p/W_i, k \in [0, W_i - 1], i \in [0, r-1]$$
$$Pr(-2, 0|r, 0, \text{Non-saturated}) = 1$$
$$Pr(-2, 0|i, 0, \text{Non-saturated}) = 1 - p, i \in [0, r-1]$$
$$Pr(-2, 1|-2, 0\ ) = 1 - \lambda$$
$$Pr(-2, 0|-2, 1) = \lambda$$
$$Pr(-2, 1|-2, 1) = 1 - \lambda$$
$$Pr(-2, 0|-1, 1) = 1 - p$$
$$Pr(-1, 1|-2, 0) = \lambda$$
$$Pr(-1, 0|-1, 1) = p$$
$$Pr(-1, 0|r, 0, \text{Saturated}) = 1$$
$$Pr(-1, 0|i, 0, \text{Saturated}) = 1 - p, i \in [0, r-1]$$

For non-saturate traffic, we have following relations

$$b_{-2,0} = (1-p)\sum_{i=0}^{r-1} b_{i,0} + b_{r,0} + \lambda b_{-2,1} + (1-p)b_{-1,1}$$
$$b_{-2,1} = (1-\lambda)b_{-2,0} + (1-\lambda)b_{-2,1}$$
$$b_{-1,0} = pb_{-1,1}$$
$$b_{-1,1} = \lambda b_{-2,0}$$

For saturated traffic, we have following relation

$$b_{-1,0} = (1-p)\sum_{i=0}^{r-1} b_{i,0} + b_{r,0} \tag{1}$$

Using Markov model in Fig. 3, we can get

$$\begin{aligned} &pb_{i-1,0} = b_{i,0},\ i \in [1,r],\ \text{i.e., } b_{i,0} = p^i b_{0,0},\ i \in [0,r] \\ &b_{i,k} = \frac{W_i - k}{(1-p)W_i} b_{i,0},\ i \in [0,r],\ k \in [1, W_i - 1] \end{aligned} \tag{2}$$

By applying the normalization condition, for non-saturated traffic, we have $\sum_{k=0}^{1}(b_{-2,k} + b_{-1,k}) + \sum_{i=0}^{r}\sum_{k=0}^{W_k-1} b_{i,k} = 1$ and for saturated traffic, we have $b_{-1,0} + \sum_{i=0}^{r}\sum_{k=0}^{W_k-1} b_{i,k} = 1$. Accordingly, we can solve for $b_{0,0}$ under non-saturated traffic condition as

$$b_{0,0} = \begin{cases} \frac{2\lambda^2(1-2p)(1-p)^2}{D_1}, & r \le m \\ \frac{2\lambda^2(1-2p)(1-p)^2}{D_2}, & r > m \end{cases} \tag{3}$$

and under saturated traffic condition as

$$b_{0,0} = \begin{cases} \frac{2p(1-2p)(1-p)^2}{D_3}, & r \le m \\ \frac{2p(1-2p)(1-p)^2}{D_4}, & r > m \end{cases} \tag{4}$$

where $D_1 = 2[1+\lambda^2(1+p)](1-p)^2(1-2p) + \lambda^2 p(1-2p)(1-p^{r+1}) + \lambda^2 p(1-p)[1-(2p)^{r+1}]W$, $D_2 = 2[1+\lambda^2(1+p)](1-p)^2(1-2p) + \lambda^2 p(1-2p)(1-p^{r+1}) + \lambda^2 p(1-p)[1-(2p)^{r+1}]W + \lambda^2 p(1-2p)2_m p^{m+1}[1-p^{r-m}]W$, $D_3 = 2(1-p)^3(1-2p) + p(1-2p)(1-p^{r+1}) + p(1-p)[1-(2p)^{r+1}]W$, and $D_4 = D_3 + p(1-2p)2^m p^{m+1}(1-p^{r-m})W$.

To transmit a data frame via backoff procedure, a WiFi device conducts the 0-th backoff with probability 1. It conducts the i-th backoff only if the previous $i$ backoffs from the 0-th backoff to the (i-1)-th backoff have failed due to the busy channel. Therefore, the probability of a WiFi device conducts the i-th backoff is $\prod_{k=0}^{i-1} b_{k,0} = \prod_{k=0}^{i-1} p^k b_{0,0}$ $(i = 1, 2, \cdots, r)$. Consider that the expected number of time slots drawn on the i-th backoff is $\frac{W_i+1}{2}$, the expected number of time slots to backoff for a TCP packet transmission is given by

$$N_{ts} = \frac{W_0+1}{2} b_{0,0} + \sum_{i=1}^{r}\prod_{k=0}^{i-1}\frac{W_i+1}{2} p^k b_{0,0}. \tag{5}$$

## V. RTT Computation Over Heterogeneous WiFi and 5G Path

RTT is a required parameter in MPTCP scheduling. IETF RFC 793 defines RTT as the elapsed time between sending a data octet and receiving an acknowledgment. We proposed a RTT computation method over heterogeneous IEEE 802.15.4 and 5G networks in [6]. However, no existing work found addresses RTT computation over heterogeneous WiFi and 5G networks. Furthermore, some works assume multiple packets can be transmitted within a RTT time period, which is not necessarily true depending on how RTT is computed. This paper uses TCP Data and TCP ACK packets to compute RTT since they are the TCP packets in data delivery. We consider a path P from data node D to data center DC with N relay nodes: $R_0 = D \to R_1 \to R_2 \to \cdots \to R_n \to R_{n+1} \to \cdots \to R_N \to DC$, where $R_N$ can be 5G node.

At a WiFi node, the time a packet consumed includes (1) random queuing time, (2) fixed DIFS time, (3) random backoff time, (4) fixed packet transmission time, (5) fixed SIFS time and (6) fixed MAC ACK transmission time. On the other hand, the time a packet spent at a 5G node only includes (1) random queuing time and (2) fixed packet transmission time. Therefore, the task is to compute random queuing time and random backoff time of WiFi node.

The expected time consumed by a TCP data packet at a WiFi node $R_n$ via immediate channel access is given by

$$T_D(R_n) = \text{DIFS} + |Data|/B_{11} + \text{SIFS} + T_{MAC-ACK}, \tag{6}$$

where $|Data|$ is the size of TCP data packet measured at WiFi PHY layer and $B_{11}$ is WiFi bandwidth.

Using the expected queuing time $T_q$ proposed in our work [6] and the expected number of backoff time slots $N_{ts}$ in Eq. (5), the expected time consumed by a TCP data packet at a WiFi node $R_n$ via random backoff is given by

$$\begin{aligned} T_D(R_n) =& T_q + \text{DIFS} + N_{ts} \times TS_{len} + |Data|/B_{11} \\ &+ \text{SIFS} + T_{MAC-ACK}, \end{aligned} \tag{7}$$

where $TS_{len}$ is WiFi time slot length.

On the other hand, the expected time spent by a TCP data packet at a 5G node $R_n$ is given by

$$T_D(R_n) = T_q + |Data|/B_{5G}, \tag{8}$$

where $B_{5G}$ is 5G bandwidth.

Therefore, the expected TCP data travel time (DTT) over N+1 hop path from data node D to DC can be computed as $DTT = \sum_{n=0}^{N} T_D(R_n)$. Denote as $R_{N+1} = DC$, the expected TCP ACK travel time (ATT) over $N+1$ hop path from DC to node D can be computed similarly as $ATT = \sum_{n=N+1}^{1} T_A(R_n)$. Therefore, the RTT over the path P is given by

$$RTT = \sum_{n=0}^{N} T_D(R_n) + \sum_{n=N+1}^{1} T_A(R_n). \tag{9}$$

## VI. Loss-Aware MPTCP Path Scheduling

Assume node D has $K$ paths $P_1, P_2, \cdots, P_K$ to DC arranged in RTT ascending order. Denote as $w_i$ the $cwnd$ of the path $P_i$. Assume TCP ACK is delayed and a new round starts after current round completes. Denote as $B_i^D$ either $B_{11}$ if node D is a WiFi node or $B_{5G}$ if node D is a 5G node on the path $P_i$. Let $T_i^j$ be the time needed to transmit $j$ data packets over path $P_i$, then $T_i^1 = DTT_i$. Consider that WiFi protocol (IEEE 802.11) specifies a transmission opportunity (TXOP) mechanism, in which once a device obtains a TXOP, it can make consecutive frame transmissions with SIFS interframe space. Therefore, travel time of $j$ data packets is given by

$$T_i^j = T_i^1 + (j-1)(\text{SIFS} + |Data|/B_i^D). \tag{10}$$

We schedule paths from the slowest path $P_K$ to the fastest path $P_1$. Our scheduling process is a tree like approach with $P_K$, $P_{K-1}$, $\cdots$, $P_1$ at the root level, first branch level, $\cdots$, (K-1)-th branch level, respectively. Recall that the first TCP data packet scheduled on $P_i$ takes $T_i^1$ time to arrive at DC. Thus, TCP data packets scheduled on path $P_{i-1}$ should arrive at DC no later than $T_i^1$ $(i = K, K-1, \cdots, 2)$. For path $P_K$, $w_k$ packets can be scheduled. For path $P_i$ $(1 \le i < K)$, $T_{i+1}^1$ is the scheduling period. We calculate number of packets can be scheduled on path $P_i$ within $T_{i+1}^1$ time. At least one packet can be scheduled since $T_i^1 \le T_{i+1}^1$. It is possible that multiple scheduling rounds can complete within $T_{i+1}^1$ time. Denote as $T_i(r)$ and $w_i(r)$ the remaining time and the $w_i$ at the start of the r-th scheduling round, respectively. The scheduling process continues until remaining time $T_i(r)$ is not enough for one packet transmission. Thus, we have following five scheduling cases at the r-th round over path $P_i$:

1) $T_i(r) < T_i^1 + ATT_i$: Has no time to transmit a new packet, scheduling ends.
2) $T_i^1 + ATT_i \le T_i(r) < T_i^{w_i(r)} + ATT_i$: Has no time to transmit $w_i(r)$ packets, but can schedule $w_i^*(r)$ packets, where $w_i^*(r) \in [1, w_i(r))$ is the largest integer satisfying $T_i^{w_i^*(r)} + ATT_i \le T_i(r)$. However, has no time for recovery or starting the (r+1)-th round. This case corresponds to the proposed CR state transmission.
3) $T_i^{w_i(r)} + ATT_i \le T_i(r) < T_i^{w_i(r)} + ATT_i + 2|ACK|/B_i^{DC} + T_i^1$: Has time to transmit $w_i(r)$ packets, but no time for recovery or starting the (r+1)-th round.
4) $T_i^{w_i(r)} + ATT_i + 2|ACK|/B_i^{DC} + T_i^1 \le T_i(r) < RTO_i + T_i^1$: Has time to complete the r-th round and start the (r+1)-th round if the lost packets can be recovered by FR, but not enough time to complete RTO retransmission. Therefore, the (r+1)-th round will not start if lost packets can not be recovered by FR. There could be three cases described in Section VI-A.
5) $T_i(r) \ge RTO_i + T_i^1$: Time is enough to complete the r-th round, retransmit lost packets via FR or RTO and start the (r+1)-th round. There could also be three cases described in Section VI-B.

### A. Case 4) Sub-Cases

- Case 4-1: No packet loss. In this case, $w_i(r)$ packets can be scheduled, the (r+1)-th will start in SS or CA or CR state with probability $p(0|w_i(r)) = \binom{w_i(r)}{0} l^0 (1-l)^{w_i(r)}$, $T_i(r+1) = T_i(r) - (T_i^{w_i(r)} + ATT_i)$, $sst_i(r+1) = sst_i(r)$,
$$w_i(r+1) = \begin{cases} 2 * w_i(r), & w_i(r) < sst_i(r) \\ w_i(r)+1, & sst_i(r) \le w_i(r) < w_i^{max} \\ w_i(r), & w_i(r) \ge w_i^{max} \end{cases}$$
- Case 4-2: With packet loss, but the number of lost packets $n \le w_i(r) - 3$ with probability $\sum_{n=1}^{w_i(r)-3} \binom{w_i(r)}{n} l^n (1-l)^{w_i(r)-n}$ so that lost packets can be recovered by FR. $w_i(r)$ packets can be scheduled, the (r+1)-th round starts in SS or CA state with probability 1, $T_i(r+1) = T_i(r) - (T_i^{w_i(r)} + ATT_i + 2|ACK|/B_i^{DC} + \sum_{n=1}^{w_i(r)-3} \binom{w_i(r)}{n} l^n (1-l)^{w_i(r)-n} T_i^n)$, $w_i(r+1) = sst_i(r+1) = \lfloor w_i(r)/2 \rfloor$.
- Case 4-3: With packet loss, but the number of lost packets $n > w_i(r) - 3$ is large enough with probability $\sum_{n=w_i(r)-2}^{w_i(r)} \binom{w_i(r)}{n} l^n (1-l)^{w_i(r)-n}$ so that lost packets can not be recovered by FR. In this case, $w_i(r)$ packets can be scheduled, but the time is not enough to trigger RTO, thus the (r+1)-th round will not start, $T_i(r+1) = 0$ and $w_i(r+1) = 0$.

### B. Case 5) Sub-Cases

- Case 5-1: Same as Case 4-1.
- Case 5-2: Same as Case 4-2.
- Case 5-3: With packet loss and the number of lost packets $n > w_i(r) - 3$ with probability $\sum_{n=w_i(r)-2}^{w_i(r)} \binom{w_i(r)}{n} l^n (1-l)^{w_i(r)-n}$ so that lost packets can be only recovered by RTO. In this case, $w_i(r)$ packets can be scheduled, the (r+1)-th round will start with probability 1. $T_i(r+1) = T_i(r) - (T_i^{w_i(r)} + RTO_i)$, $sst_i(r+1) = \lfloor w_i(r)/2 \rfloor$ and $w_i(r+1) = w_i^{min}$.

In summary of sub-cases VI-A and VI-B, given the time $T_i(r)$, the recursive process can go through all three state transition scenarios with corresponding probability.

## VII. Performance Evaluation

This section presents performance evaluation of proposed MPTCP techniques under varying network configurations.

### A. Simulation Settings

We used NS3 simulator with IEEE 802.11n and LTE communication protocols. The 802.11n bandwidth is set to 54 $Mbps$ with 20 MHz channel and LTE bandwidth is set to 100 $Mbps$ with 20 MHz channel. For data traffic, each data node delivers 100 packets to DC and each packet is 1000 bytes generated using Poison process with $\lambda = 10/s$. Standard IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) is applied for multipath discovery with the maximum number of paths $K = 3$. We set $1 \le cwnd_{min} \le 5$, $10 \le cwnd_{max} \le 20$, $sst_{start} = 16$, $20s \le RTO\ Timer \le 50s$, infinite queue size for data nodes and $rwnd = \infty$ for DC. In the simulation, collision probability $p$ at each data node and path loss rate $l$ are measured and applied in scheduling. The node deployment scenarios emulate IoT applications such as smart

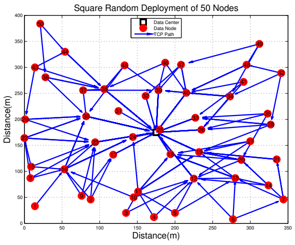


Fig. 4: Random Deployment of 50 Data Nodes with DC at Center

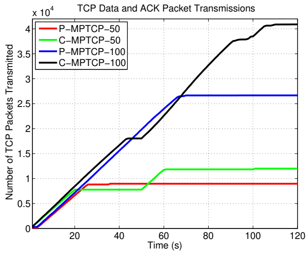


Fig. 5: Number of TCP Data and ACK Packet transmissions

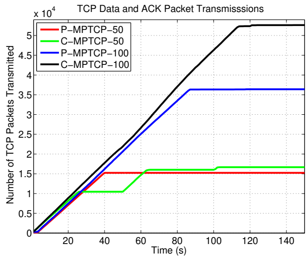


Fig. 8: Number of TCP Packet Transmissions

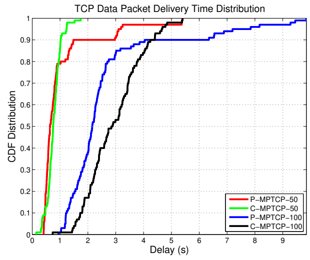


Fig. 9: TCP Data Packet Delay Distribution

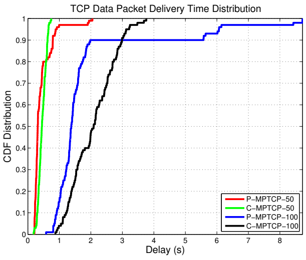


Fig. 6: TCP Data Packet Latency Distribution

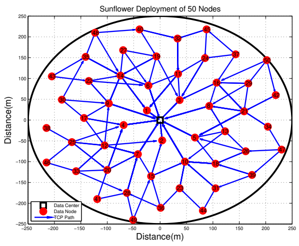


Fig. 7: Sunflower Deployment of 50 Data Nodes with DC at Center

meter, smart agriculture and smart factory. We evaluated the proposed MPTCP (P-MPTCP) techniques in four aspects: (i) Number of TCP data and ACK packet transmissions, (ii) TCP data packet latency, (iii) TCP data throughput, and (iv) TCP data packet delivery rate, where data packet delivery rate is measured as the number of TCP data packets received by DC or relay data nodes divided by the number of TCP data packets transmitted or relayed, and data packet latency is measured as the time difference between the time a data packet is transmitted by source data node and the time that data packet is received by DC. We also measured data delivery completion time. Conventional MPTCP (C-MPTCP) with the MinRTT scheduler and standard NewReno algorithm is used as the baseline.

*B. Square Node Deployment with DC at Center*

In this case, all data nodes are WiFi nodes. DC is placed at center of deployment area with 50 and 100 data nodes randomly deployed in a 350m×400m rectangle and a 500m×500m square, respectively. Fig. 4 demonstrates the node deployment and MPTCP paths for 50 data node case.

Fig. 5 shows the number of TCP data and ACK packet transmissions. P-MPTCP reduces C-MPTCP packet transmissions from 11998 to 8957 by 25.3% and 40897 to 26665 by 34.8% for 50 and 100 data nodes, respectively.

Both P-MPTCP and C-MPTCP achieve over 99.8% of overall TCP data packet delivery rate. However, for 100 data nodes, C-MPTCP delivery rate drops to 88% at the beginning, indicating P-MPTCP is more stable.

Fig. 6 shows CDF distribution of TCP data packet latency. P-MPTCP delivers 80% and 90% data packets faster than C-MPTCP does for 50 and 100 data nodes, respectively. In terms of completion time, C-MPTCP takes 102s and 113s to deliver 5000 and 10000 data packets, respectively. P-MPTCP shortens these times to 36s by 64.7% and 70s by 38.1%, respectively, indicating P-MPTCP is more efficient.

For data throughput, P-MPTCP improves C-MPTCP throughput from 392.2 kbps to 1111.1 kbps by 183.3% and 709.8 kbps to 1142.9 kbps by 61.0% for 50 and 100 data nodes, respectively, indicating P-MPTCP is more efficient.

*C. Sunflower Node Deployment with DC at Center*

In this case, all data nodes are WiFi nodes. 50 and 100 data nodes are deployed using Sunflower deployment algorithm within circles of 250m and 350m radius, respectively, with DC at center. These node deployments are sparser than the corresponding square deployments. Fig. 7 shows deployment of 50 data nodes and MPTCP paths.

Fig. 8 shows the number of TCP packet transmissions. For 50 nodes, C-MPTCP transmits 16669 packets and P-MPTCP transmits 15246 packets, a 8.5% of transmission reduction. For 100 nodes, C-MPTCP transmits 52650 packets and P-MPTCP transmits 36391 packets, a 30.9% of transmission reduction. Compared to square node deployments, more packets are transmitted due to longer paths caused by the sparser node deployment. These results reveal that longer paths have more impact than higher interference does.

Both P-MPTCP and C-MPTCP achieve over 99.7% of overall TCP data packet delivery rate. However, for 100 node deployment, C-MPTCP delivery rate drops to 93.5% at the beginning, indicating P-MPTCP is more stable.

Fig. 9 illustrates data packet latency. For data packet latency, P-MPTCP delivers 80% and 90% data packets faster than C-MPTCP does for 50 and 100 nodes, respectively. In terms of completion time, C-MPTCP takes 104s and 124s to deliver 5000 and 10000 data packets, respectively. On the other hand, P-MPTCP takes 40s, a 61.5% of latency reduction, and 84s, a 32.2% of latency reduction, respectively.

For data throughput, P-MPTCP improves C-MPTCP throughput from 385.0 kbps to 1000.0 kbps by 159.7% and 645.0 kbps to 952.0 kbps by 47.6%, respectively.

*D. Square Node Deployment with DC at Corner*

This simulation setting is to evaluate performance of P-MPTCP with more congested bottlenecks and most impor-

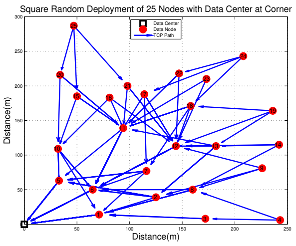


Fig. 10: Random Deployment of 25 Nodes without 5G Node

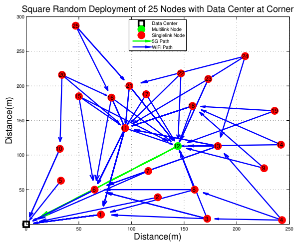


Fig. 11: Random Deployment of 25 Nodes with One 5G Node

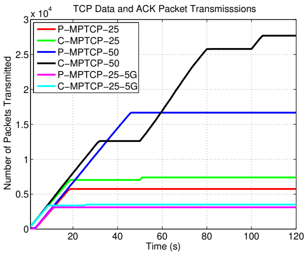


Fig. 12: Number of TCP Packet Transmissions

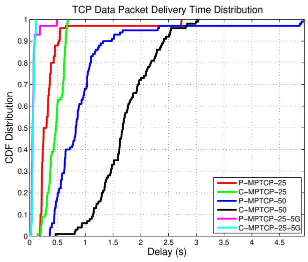


Fig. 13: TCP Data Packet Delay Distribution

tantly, to demonstrate the impact of 5G node. DC is placed at corner, 25 and 50 data nodes are randomly deployed in a 250m×250m square and a 350m×400m rectangle, respectively. Figs. 10 and 11 illustrate network topology and MPTCP paths for 25 nodes without and with 5G node, respectively. Fig. 11 shows that neighboring WiFi nodes discover 5G node 12, which has a direct path to DC, and build MPTCP paths through the 5G node.

Fig. 12 shows the number of TCP packet transmissions. For 25 and 50 nodes without 5G node, C-MPTCP transmits 7395 and 27690 packets, respectively. P-MPTCP reduces transmissions to 5746 by 22.2% and 16669 by 39.8%, respectively. For 25 nodes with one 5G node, C-MPTCP and P-MPTCP reduce their transmissions to 3509 by 52.5% and 3118 by 45.7%, respectively. These results emphasize the impact of 5G node on both C-MPTCP and P-MPTCP. With one 5G node, P-MPTCP transmits 11% less packets than C-MPTCP does. For 50 nodes without 5G node, this deployment takes more transmissions to deliver 5000 data packets than 50 nodes do with DC at center. It reveals the impact of traffic congestion at bottlenecks and longer paths.

For 25 nodes with and without 5G node, both C-MPTCP and P-MPTCP achieve 100% of data packet delivery rate. For 50 nodes without 5G node, C-MPTCP achieves 96.6% of data packet delivery rate and P-MPTCP achieves 99.8% of data packet delivery rate, a 3.2% of improvement.

Fig. 13 shows data packet latency. For 25 and 50 nodes without 5G node, P-MPTCP delivers 95% of data packets much faster than C-MPTCP does. For 25 nodes, one 5G node significantly improves delay distribution for both C-MPTCP and P-MPTCP. In terms of completions time, without 5G node, C-MPTCP takes 51s and 105s to deliver 2500 and 5000 data packets, respectively, and P-MPTCP takes 19s and 47s, a 62.7% and a 55.2% of latency reduction, respectively. For 25 nodes with one 5G node, C-MPTCP and P-MPTCP reduce their completion time to 26s by 68.6% and 12s by 38.8%, respectively. In addition, P-MPTCP improves C-MPTCP completion time by 53.8%. These results reveal the impact of 5G node.

For 25 nodes without 5G node, P-MPTCP improves C-MPTCP data throughput from 392 kbps to 1052 kbps by 168.4%. For 25 nodes with one 5G node, C-MPTCP and P-MPTCP improve their throughput to 769 kbps by 96.2% and 1667 kbps by 241.6%, respectively. In addition, P-MPTCP improves C-MPTCP throughput by 116.8%. For 50 nodes without 5G node, P-MPTCP improves C-MPTCP data throughput from 381 kbps to 851 kbps by 123.4%.

## VIII. Conclusion

This paper studies MPTCP over heterogeneous WiFi and 5G networks. It models a proposed 5-state congestion control algorithm and fully functional WiFi CSMA function as Markov chains. An innovative RTT computation method is introduced for novel loss-aware data transmission scheduling over multiple paths. Compared with conventional MPTCP, the proposed MPTCP techniques can reduce up to 52.5% TCP packet transmission, shorten up to 61.5% data delivery time, improve up to 183.3% data throughput and increase up to 3.2% data delivery rate. In addition, 5G node can significantly improve network performance, path length and node deployment can also impact network performance.

## References


[1] K. Nadeem and T. Jadoon, "An ns-3 MPTCP Implementation," in *14th EAI International Conference on Quality, Reliability, Security and Robustness in Heterogeneous Networks*, 2018.

[2] N. Kuhn, E. Mifdaoui, A. Sarwar, O. Mehani, and O. Boreli, "DAPS: Intelligent delay-aware packet scheduling for multipath transport," in *IEEE International Conference on Communications (ICC)*, 2014.

[3] S. Ferlin, O. Alay, and et al, "Blest: Blocking estimation-based mptcp scheduler for heterogeneous networks," in *International Federation for Information Processing (IFIP) Networking*, 2016.

[4] W. Yang, P. Dong, and et al, "Loss-Aware Throughput Estimation Scheduler for Multi-Path TCP in Heterogeneous Wireless Networks," *IEEE Transactions on Wireless Communications*, vol. 20, 2021.

[5] D. Weber, C. Fuchs, N. Auler, B. Schütz, and N. Aschenbruck, "Multipath TCP Scheduling in the Age of Buffer Bloat Optimizations," in *IEEE 48th Conference on Local Computer Networks (LCN)*, 2023.

[6] J. Guo, K. Parsons, Y. Nagai, T. Sumi, N. Sakaguchi, H. Tsuchida, P. Wang, and P. Orlik, "Multipath TCP Over Multi-Hop Heterogeneous Wireless IoT Networks," in *IEEE International Conference on Communications (ICC)*, 2024.

[7] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas in Communications*, vol. 18, 2000.

[8] M. Ergen and P. Varaiya, "Throughput Analysis and Admission Control for IEEE 802.11a," *Mobile Networks and Applications*, vol. 10, pp. 705–716, 2005.

[9] Q. Wu, H. Zhang, Z. Li, Y. Liu, and C. Zhang, "Performance evaluation of the V2I fair access with a finite retry limit," *EURASIP Journal on Wireless Communications and Networking*, vol. 20, 2018.

[10] C. Casctti and M. Meo, "A New Approach to Model the Stationary Behavior of TCP Connections," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2000.