Multipath TCP Over Multi-Hop Heterogeneous Wireless IoT Networks

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

TR2024-071 June 11, 2024

Abstract

With the advent of 5G and beyond communication technologies, the consumer IoT devices are evolving from cur- rent generation to next generation. Next generation IoT devices can support multiple communication interfaces and perform more functions. Accordingly, IoT network technologies must adapt to the emerging multi-link devices to improve network performance. Multipath TCP (MPTCP) is desired for networks with multi-link devices and has achieved success in computer networks. However, MPTCP has not been well studied for wireless networks. To that end, this paper presents MPTCP techniques for heterogeneous wireless IoT networks consisting of IEEE 802.15.4 nodes and 5G nodes. We propose a path builder, an adaptive congestion controller and an innovative path scheduler. We evaluated our MPTCP techniques under varying network configurations. Compared with conventional MPTCP, the proposed MPTCP can significantly reduce the number of packet transmissions, shorten packet delivery time, improve network throughput and packet delivery rate.

IEEE International Conference on Communications (ICC) 2024

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

Multipath TCP Over Multi-Hop Heterogeneous Wireless IoT Networks

Jianlin Guo*, Kieran Parsons*, Yukimasa Nagai†, Takenori Sumi†, Naotaka Sakaguchi†, Hikaru Tsuchida†,
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*—With the advent of 5G and beyond communication technologies, the consumer IoT devices are evolving from current generation to next generation. Next generation IoT devices can support multiple communication interfaces and perform more functions. Accordingly, IoT network technologies must adapt to the emerging multi-link devices to improve network performance. Multipath TCP (MPTCP) is desired for networks with multi-link devices and has achieved success in computer networks. However, MPTCP has not been well studied for wireless networks. To that end, this paper presents MPTCP techniques for heterogeneous wireless IoT networks consisting of IEEE 802.15.4 nodes and 5G nodes. We propose a path builder, an adaptive congestion controller and an innovative path scheduler. We evaluated our MPTCP techniques under varying network configurations. Compared with conventional MPTCP, the proposed MPTCP can significantly reduce the number of packet transmissions, shorten packet delivery time, improve network throughput and packet delivery rate.**

***Index Terms*—Multipath TCP, round trip time, path scheduling, path construction, congestion control, IoT networks.**

I. Introduction

The consumer IoT devices are evolving from current generation to next generation. However, it is impractical to completely remove the deployed current generation devices. Thus, IoT networks will consist of the mixed current and next generation devices. Take smart meter network for example, current generation meters support one communication interface and collect regular metering data only, but next generation meters can support multiple communication interfaces, collect regular metering data and sense power supply information, which is critical for power suppliers to make predictive maintenance and diagnose the cause of the abnormal events and must be reliably delivered. Accordingly, power supply information can be delivered using MPTCP over multiple paths for more reliable and faster delivery.

MPTCP specified in IETF RFC 8684 is an evolution of TCP to allow the simultaneous use of multiple communication interfaces for efficient data delivery. MPTCP aims to improve data delivery reliability, improve data throughput and reduce data latency via multiple paths. MPTCH achieves success in computer networks. The studies have shown that simultaneously using multiple communication interfaces can achieve higher throughput and complete transmissions in a shorter time. The main components of MPTCP include path management, path scheduling and congestion control.

Despite the success of the MPTCP in computer networks, its deployment over wireless networks is not well studied, especially over the carrier sense multiple access (CSMA) based wireless networks, in which random backoff delay incurs great challenges for path scheduling.

This paper studies MPTCP over multi-hop heterogeneous wireless networks consist of IEEE 802.15.4 nodes and 5G nodes. We propose a path construction method, an adaptive congestion control algorithm and a novel path scheduling mechanism for MPTCP to fit wireless IoT networks better. We evaluated the proposed MPTCP techniques under varying network configurations by using NS3 simulator and observed interesting insights. To the best of our knowledge, we are the first to study MPTCP over CSMA based wireless networks.

The rest of this paper is organized as follows. Section II presents related works. Section III provides MPTCP path construction. Section IV introduces adaptive congestion control. The Markov chain modeling of the IEEE 802.15.4 CSMA algorithm is presented in Section V. MPTCP path scheduling is provided in Section VII. Performance evaluation is conducted in Section VIII. Finally, we conclude our paper in Section IX.

II. Related Works

For the MPTCP path management, the Linux Kernel [1] implements four path managers: default, fullmesh, ndiffPorts and binder. Recently, researchers have proposed path management methods for wireless networks, e.g., article [2] proposes a cross-layer path management approach for heterogeneous vehicular networks. However, no existing work found addresses the MPTCP path construction.

Path scheduling is the most studied MPTCP component. The fastest round trip time (Fastest-RTT) is a default scheduler, in which the paths are scheduled based on RTT with the smaller RTT paths having higher priorities. There are scheduling methods that enhances the Fastest-RTT scheduler by considering other metrics. Authors in [3] propose a delay-aware packet scheduling (DAPS) for MPTCP. The DAPS aims to reduce the receiver's buffer blocking time considered as a main parameter to enhance the QoS in wireless environments. Work [4] presents a blocking estimation-based MPTCP scheduler for heterogeneous networks to minimize head of line blocking. Paper [5] proposes a loss-aware throughput estimation scheduler for MPTCP in heterogeneous wireless networks and a method to compute the number of packets that can be transmitted over a path in a scheduling round. However, these studies do not address the RTT computation, which is required by the MPTCP path scheduling and challenging to compute in wireless networks, especially in the CSMA based wireless IoT networks. In addition, these works use pre-configured network topology for performance evaluation without considering network dynamics such as node location, node connectivity and link quality. Wireless networks are dynamic with characteristics such as link quality and packet loss varying dynamically. Accordingly, MPTCP path scheduling over wireless IoT networks needs to take wireless dynamics into account.

The traffic congestion control is critical for MPTCP to achieve high network efficiency especially in wireless networks. Although there are alternative congestion control methods, the NewReno algorithm is a default congestion controller for MPTCP specified in IETF RFC 6582.

This paper proposes MPTCP techniques for heterogeneous IoT networks consisting of a data center (DC), IEEE 802.15.4 data nodes and data nodes with both IEEE 802.15.4 and 5G communication interfaces referred to as multi-link (ML) nodes. Consider that D2D communication in 5G is not yet fully supported, we assume that ML nodes can communicate with IEEE 802.15.4 nodes, 5G base stations (BSs) and DC, which is considered as a ML node.

III. Build MPTCP Paths Over Heterogeneous Wireless IoT Networks

In wired networks, paths are built via physical wires even more logical paths can be established. In the CSMA based wireless networks, nodes form a mesh topology based on physical communication links. A node may have connectivity with many nodes and thus, can establish a large number of paths to a destination. It is impractical to build too many paths. Accordingly, we define a number of path threshold NP_{thr} to limit the number of paths to be established.

This paper proposes a path construction method for heterogeneous wireless IoT networks. For ML nodes, paths in 5G network are managed by BS network. Thus, we can conceptually build 1-hop or 2-hop path depending on if ML nodes connect to DC directly or via BS network. For 802.15.4 nodes, we extend IETF RPL, a multi-path routing protocol, to build paths. RPL organizes nodes in a network as a Destination Oriented Directed Acyclic Graph (DODAG) using DIO message to establish upward routes and DAO message to setup downward routes. We extend DIO to contain path traversed and node type (NT), 0 for 802.15.4 node and 1 for ML node, and extend DAO to contain path built and path ID. The extended fields in DIO message are used by downstream nodes to build paths. The extended fields in DAO message are used by upstream nodes to store paths for downstream nodes as {Source Node ID, Path ID, Upward Next Hop, Downward Next Hop}.

Node DC starts path establishment by broadcasting DIO message via both 5G and IEEE 802.15.4 interfaces with path = {DC} and NT = 1. Upon receiving a DIO message over 5G network, 5G BSs rebroadcast the received DIO message and a ML node builds path = {Node, DC} or {Node, BS, DC} depending on transmitter of DIO message, updates DIO message with the path built and NT = 1, broadcasts the updated DIO message in 802.15.4 network to propagate path establishment, assigns a path ID and sends a DAO message to DC. Upon receiving the DIO messages, an 802.15.4 node selects parents using RPL protocol criteria such as rank and builds path = {Node, Path contained in the received DIO message}. If the number of paths exceeds NP_{thr}, the node can replace an existing path with a better path by considering path length and the number of ML nodes on the path. For each path built, an 802.15.4 node broadcasts an updated DIO message with the path built and NT = 0 to propagate path establishment. The node then assigns a path ID and sends a DAO message for upstream nodes to store its path.

IV. Adaptive NewReno Algorithm for Wireless IoT Networks

MPTCP NewReno algorithm uses congestion window (*cwnd*), slow start threshold (*sst*) and receiver window (*rwnd*) to control traffic congestion. The *cwnd* limits the number of packets can be transmitted in a scheduling round and the *rwnd* indicates amount of data receiver is willing to accept. To describe the NewReno algorithm, denote as w the *cwnd* for short. The NewReno algorithm starts in slow start (SS) state with $w = w_{min}$ 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 increments 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 goes to FR state, both sst and w are set to $w/2$. If algorithm goes to RTO state, sst is set to $w/2$ and w is then set to w_{min}.

NewReno algorithm was designed for computer networks with relatively stable network environment. However, wireless IoT networks are dynamic. For multi-hop data-centric IoT networks, the bottlenecks are the nodes close to data center. Therefore, this paper proposes an adaptive NewReno (A-NewReno) algorithm to enhance NewReno algorithm for MPTCP over multi-hop wireless IoT networks. A-NewReno algorithm provides following adaptations: (1) The w_{min} and w_{max} adaptation: The w_{min} and w_{max} are not uniform across network. Data nodes close to data center have smaller w_{min} and larger w_{max}. As data nodes get away from data center, w_{min} becomes larger and w_{max} becomes smaller. This enhancement considers factor that it is time consuming for data nodes away from data center to recover packet loss due to multi-hop relay and therefore, a relatively stable w is needed. On the other hand, data nodes close to data center can quickly respond to packet loss. (2) RTO timer adaptation: Setting RTO timer is challenging. IETF RFC 793 provides a method to set RTO = min{UB, max{LB,(β*SRTT)}}, where UB is an upper bound (e.g., 1 minute), LB is a lower bound (e.g., 1 second), β is a delay variance factor and smoothed RTT (SRTT) is given by SRTT = α*SRTT + (1- α)*RTT, where α is a smoothing factor. In this paper, RTO timer is proportional to path length since longer paths typically take more time to deliver data. (3) The w update frequency adaptation: It is not necessary for data nodes away from data center to update w in each scheduling round. These nodes can explore w values that provide good performance and then maintain w until the w leads to poor performance.

Denote as l the packet loss probability and $p(x|w)$ the probability of x packet loss in w packets. Assume packet losses are independent from each other, then $p(x|w)$ obeys the Bernoulli distribution, i.e., $p(x|w) = \binom{x}{w} l^x (1-l)^{w-x}$. Consider that the FR requires $w \geq 4$ and refer to work [6], we can get the state transition probabilities as: $p_{SS} = p_{CA} = p(0|w)$, $p_{RTO} =$

$$\begin{cases} \sum_{i=1}^{w-3} p(i|w)(1-(1-l)^i) + \sum_{i=w-2}^{w} p(i|w), & w \geq 4 \\ 1 - p(0|w), & w < 4 \end{cases}$$

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

V. Modeling IEEE 802.15.4 Non-Slotted CSMA Algorithm

IEEE 802.15.4 random backoff delay can be significant. Accordingly, to compute the RTT over a path consisting of IEEE 802.15.4 node, the random backoff delay must be considered. IEEE 802.15.4 standard specifies two CSMA operation modes: Slotted and Non-Slotted. Papers [7] and [8] model IEEE 802.15.4 Slotted CSMA algorithm as chain model and Markov chain model, respectively. Work [9] models 802.15.4 Non-Slotted CSMA algorithm as a chain model, but not a Markov chain model. Authors in [10] aim to model 802.15.4 Non-Slotted CSMA algorithm as a Markov chain model. However, the model incorporates backoff counter decrement process that is not a memoryless process as its value depends on its history [9]. Accordingly, the model in paper [10] is not a Markov chain model.

IoT networks typically adopt IEEE 802.15.4 Non-Slotted CSMA mode. Thus, we model IEEE 802.15.4 Non-Slotted CSMA algorithm as a Markov chain model to compute the RTT. IEEE 802.15.4 Non-Slotted CSMA algorithm uses parameters *macMinBe*, *macMaxBe* and *macMaxCsmaBackoffs*, which is denoted as NB_{max} in this paper. The backoff window W_i on the i-th backoff is given by $W_i = min\{2^{macMinBe+i}, 2^{macMaxBe}\}$, $i = 0, 1, 2, \cdots, NB_{max}$. On the i-th backoff, node has a uniform probability $\frac{1}{W_i}$ to draw $0, 1, \cdots, W_i - 1$ backoff periods. Therefore, $\sum_{i=0}^{NB_{max}} (W_i - 1)$ is the maximum number of backoff periods that can be backoffed in a channel access attempt. Consider that in IEEE 802.15.4 Non-Slotted CSMA mode, one CCA is performed after completion of each backoff, $T_{max} = \sum_{i=0}^{NB_{max}} W_i$ is the maximum number of backoff periods that can be consumed in a channel access attempt. We divide time into the unit of backoff period and assume channel access contention starts at backoff period 1. On the i-th backoff, a node has a uniform probability $\frac{1}{W_i}$ to draw $0, 1, \cdots, W_i - 1$ backoff periods. Therefore, on the i-th backoff, the expected number of backoff periods is given by $\frac{W_i - 1}{2}$ $(i = 0, 1, 2, \cdots, NB_{max})$. Consider that in IEEE 802.15.4 Non-Slotted CSMA mode, only one CCA is performed at the expected $\frac{W_i+1}{2}$-th backoff period, we denote as k_i the expected number of backoff periods elapsed up to the i-th backoff

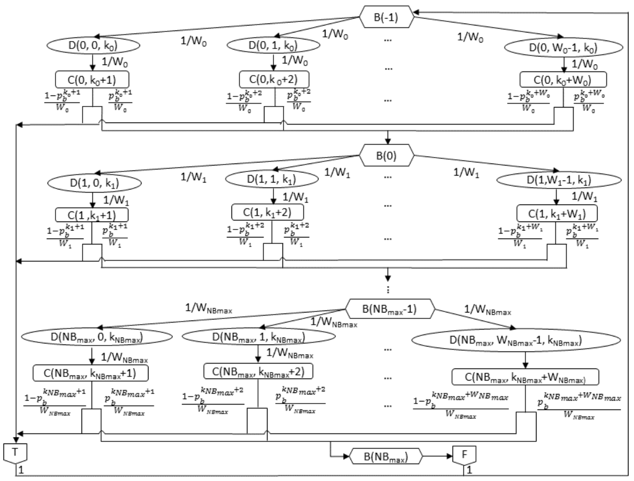

Fig. 1. Proposed Markov Chain Model for IEEE 802.15.4 Non-Slotted CSMA Algorithm

$$k_0 = 0\ , k_i = \sum_{t=0}^{i-1} \frac{W_t + 1}{2},\ i = 1, 2, ..., NB_{max}. \qquad (1)$$

To model IEEE 802.15.4 Non-Slotted CSMA algorithm as a Markov chain model, we omit backoff counter decrement states since the state transition probability in backoff counter decrement process is always 1. We define following Markov chain states

- D(i,j,k_i): Node performs the i-th backoff by delaying j backoff periods starting at the k_i-th backoff period, $0 \leq i \leq NB_{max}$, $0 \leq j \leq W_i - 1$ and $i \leq k_i \leq T_{max}$.
- C(i,j): Node performs CCA at the (k_i+j+1)-th backoff period on the i-th backoff, $0 \leq i \leq NB_{max}$ and $0 \leq j \leq W_i - 1$.
- B(-1): Node prepares to gain channel for a data packet transmission. To transmit a data packet, IEEE 802.15.4 CSMA algorithm performs first backoff, i.e., 0-th backoff, no matter channel is idle or not. Therefore, the channel is considered as busy on the (-1)-th backoff.
- B(i): Channel is busy on the i-th backoff, $0 \leq i \leq NB_{max}$.
- T: Node gains channel and starts packet transmission.
- F: Node fails gaining the channel as the number of backoffs reaches the NB_{max}.

Denote as p_b^k the channel busy probability in the backoff period k, $1 \leq k \leq T_{max}$. Fig. 1 shows the proposed Markov chain mode for IEEE 802.15.4 Non-Slotted CSMA algorithm, where hexagon represents state B(i), circle serves as state D(i,j,k), rectangle shows state C(i,j), pentagon with caption T acts as state T and pentagon with caption F illustrates state F. Define following state transition probabilities

$$\begin{aligned}
&d_{i,j} = p(D(i,j,k_i)|B(i-1)), 0 \leq i \leq NB_{max}, \\
&\quad 0 \leq j \leq W_i - 1 \\
&c_{i,j} = p(C(i,k_i+j+1)|D(i,j,k_i), 0 \leq i \leq NB_{max}, \\
&\quad 0 \leq j \leq W_i - 1 \qquad (2) \\
&b_{i,j} = p(B(i)|C(i,k_i+j), 0 \leq i \leq NB_{max}, 1 \leq j \leq W_i \\
&t_{i,j} = p(T|C(i,k_i+j)), 0 \leq i \leq NB_{max}, 1 \leq j \leq W_i \\
&f_j = p(F|C(NB_{max}, k_{NB_{max}}+j)), 1 \leq j \leq W_{NBmax}
\end{aligned}$$

Define $b_{-1} = 1$ and denote as b_i the probability $p(B_i)$ ($0 \leq i \leq NB_{max}$). Using the Markov chain model in Fig. 1, we can get following equations

$$\begin{aligned}
&d_{i,j} = \frac{b_{i-1}}{W_i}, 0 \leq i \leq NB_{max}, 0 \leq j \leq W_i - 1 \\
&c_{i,j} = d_{i,j}, 0 \leq i \leq NB_{max}, 1 \leq j \leq W_i - 1 \\
&b_{i,j} = d_{i,j} p_b^{k_i+j}, 0 \leq i \leq NB_{max}, 1 \leq j \leq W_i \\
&b_i = \sum_{j=1}^{W_i} b_{i,j}, 0 \leq i \leq NB_{max} \qquad (3) \\
&t_{i,j} = d_{i,k_i+j}(1 - p_b^{k_i+j}), 0 \leq i \leq NB_{max}, 1 \leq j \leq W_i \\
&f_j = d_{NB_{max},k_{NB_{max}}+j} p_b^{k_{NB_{max}}+j}, 1 \leq j \leq W_{NBmax} \\
&p(T) = \sum_{i=0}^{NB_{max}} \sum_{j=1}^{W_i} t_{i,j}, \quad p(F) = \sum_{j=1}^{W_{NB_{max}}} f_j \\
&p(C_0|T) = 1, \quad p(C_0|F) = 1
\end{aligned}$$

To transmit a data packet, an 802.15.4 node 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 failed. Therefore, the probability of an 802.15.4 node conducts the i-th backoff is $\prod_{n=0}^{i-1} b_n$ ($i = 1, 2, \cdots, NB_{max}$). Consider that the expected number of backoff periods consumed on the i-th backoff is $\frac{W_i+1}{2}$, the expected number of backoff periods to gain channel for a TCP packet transmission is given by

$$N_{bp} = \frac{W_0 + 1}{2} + \sum_{i=1}^{NB_{max}} \prod_{n=0}^{i-1} b_n \frac{W_i + 1}{2}. \qquad (4)$$

VI. RTT Computation Over Heterogeneous Path

MPTCP scheduling depends on the RTT, which is defined by IETF RFC 793 as the elapsed time between sending a data octet and receiving an acknowledgment. However, no existing work found addresses RTT computation. This paper uses TCP SYN and TCP ACK messages to compute RTT since both messages do not have payload. We consider a path from data node D to data center DC with N relay nodes: $R_0 = D \rightarrow R_1 \rightarrow R_2 \rightarrow \cdots \rightarrow R_n \rightarrow R_{n+1} \rightarrow \cdots \rightarrow R_N \rightarrow DC$, where R_{N-1} and R_N can be ML nodes.

At an 802.15.4 node, the time a packet consumed includes (1) random queuing time, (2) random channel access delay, (3) fixed RX to TX turnaround time, (4) fixed packet transmission time (once packet size and bandwidth are given) and (5) fixed MAC ACK transmission time (802.15.4 MAC layer sends MAC ACK before forwarding TCP packet to upper layers). However, the time a packet spent at a ML node only includes (1) random queuing time and (2) fixed packet transmission time. Therefore, the task is to compute queuing time and channel access delay of 802.15.4 node. It is impractical to compute the exact value of a random variable. Accordingly, we compute the expected values.

A. Expected Queuing Time

We use a M/M/1/K queue to model the expected queuing time. Assume each node has one queue of size K with a single server and packet arrives according to a Poisson process with rate λ (packets/s). Service process follows an exponential distribution with rate μ (packets/s). Let B_{15} and B_{5g} be 802.15.4 bandwidth and 5G bandwidth, respectively, then $\mu = \frac{B_{15}}{8*PacketSize}$ and $\frac{B_{5g}}{8*PacketSize}$, respectively. Denote as $\rho = \frac{\lambda}{\mu}$. From M/M/1/K theory, the probability that queue contains n (n = 0,1,2,$\cdots$,K) packets is

$$p_n = \begin{cases} 1/(K+1), & \rho = 1 \\ (1-\rho)\rho^n/(1-\rho^{K+1}), & \rho \neq 1 \end{cases} \qquad (5)$$

Let N_q be the expected number of packets in the queue, the N_q can be calculated as

$$N_q = \begin{cases} K/2, & \rho = 1 \\ \rho/(1-\rho) - (K+1)/(1-\rho^{K+1}), & \rho \neq 1 \end{cases} \qquad (6)$$

Assume queue is not full (otherwise packet is discarded), then considering current packet being added into the queue, the expected queuing time T_q is given by

$$T_q = (N_q + 1)/\mu. \qquad (7)$$

B. Expected RTT Over A Heterogeneous Path

Using T_q in Eq. (7) and N_{bp} in Eq. (4), the expected time a packet consumed at an 802.15.4 node R_n is given by

$$\begin{aligned} T_e(n) = T_q + N_{bp} * |BP| + T_{turnaround} \\ + |SYN|/B_{15} + T_{MAC-ACK}, \end{aligned} \qquad (8)$$

where $|BP|$ is the length of the backoff period and $|SYN|$ is the size of TCP SYN packet measured at PHY layer, which is also the TCP packet header size (HS). On the other hand, the expected time a packet spent at a ML node R_n is

$$T_e(n) = T_q + |SYN|/B_{5g}. \qquad (9)$$

Therefore, TCP SYN packet travel time (STT) over N+1 hop path from node D to data center DC is given by

$$STT = \sum_{n=0}^{N} T_e(n). \qquad (10)$$

Since SYN and ACK packets have same size, we can assume ACK packet travel time (ATT) from DC to node D is same as STT. Therefore, the RTT over the given path is given by

$$RTT = STT + ATT = 2 * \sum_{i=0}^{N} T(i). \qquad (11)$$

VII. MPTCP Path Scheduling Over Multi-Hop Heterogeneous IoT Networks

MPTCP scheduling is to schedule TCP packets over multiple paths for in-order arrival at destination to avoid head-of-line blocking and full buffer packet drop. Assume node D has NP_{thr} paths $P_1, P_2, \cdots, P_{NP_{thr}}$ arranged in RTT ascending order. Consider a path P_i ($i = 1, 2, \cdots, NP_{thr}$) and denote as w_i the *cwnd* of the path P_i. Assume TCP ACK is delayed for w_i packets and a new round starts after current round completes. Denote as B_i either B_{15} if node D is an 802.15.4 node or B_{5g} if node D is a ML node on the path P_i. Scheduling is conducted based on the fastest moving average RTT. We calculate the number of packets can be scheduled on each path. For path $P_{NP_{thr}}$, $w_{NP_{thr}}$ packets can be scheduled. For path P_i ($i = 1, 2, \cdots, NP_{thr} - 1$), denote as $T_i(1) = RTT_{i+1}/2$, we compute the number of packets that can be scheduled in time period $T_i(1)$. Multiple rounds can complete in $T_i(1)$ time. Denote as $T_i(r)$ and $w_i(r)$ the remaining time and the w_i at the start of r-th round, respectively. To transmit m TCP data packets with payload of size PS over path P_i, we replace $|SYN|$ with HS+PS in Eq. (10) to get the expected time to deliver first data packet as $T_i^1 = RTT_i/2 + \sum_{n=0}^{N} PS/B_i$. The remaining $m-1$ data packets can be transmitted sequentially, i.e., once one packet is transmitted, node D starts channel access contention for next packet transmission. Therefore, $T_i^m = T_i^1 + (m-1)(T_e(D) + (HS+PS)/B_i)$ is the expected time to deliver m data packets over path P_i. Extending work [5] to consider TCP packet transmission delay time, we have following four cases for the r-th scheduling round:

1) $T_i(r) < T_i^1$: Has no time to transmit a new packet, scheduling ends.
2) $T_i^1 \leq T_i(r) < T_i^{w_i(r)} + RTT_i/2 + T_i^1$: Has time to transmit $w_i(r)$ packets, but no time for recovery or starting (r+1)-th round.
3) $T_i^{w_i(r)} + RTT_i/2 + T_i^1 \leq T_i(r) < RTO_i + RTT_i/2$: Has time to complete r-th round and start (r+1)-th round if lost packets can be recovered by FR, but has no 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 VII-A.
4) $T_i(r) \geq RTO_i + RTT_i/2$: Time is enough to finish r-th round, retransmit lost packets via FR or RTO and start (r+1)-th round. There could also be three cases described in Section VII-B.

A. Case 3 Sub-Cases

- Case 3-1: No packet loss. In this case, $w_i(r)$ packets can be scheduled, the (r+1)-th will start in SS or CA state with probability $p(0|w_i(r)) = \binom{0}{w_i(r)} l^0 (1-l)^{w_i(r)}$, $T_i(r+1) = T_i(r) - T_i^{w_i(r)}$, $sst_i(r+1) = sst_i(r)$ and
$$w_i(r+1) = \begin{cases} 2 * w_i(r), & w_i(r) < sst_i(r) \\ w_i(r) + 1, & w_i(r) \geq sst_i(r) \end{cases} \tag{12}$$
- Case 3-2: With packet loss, but the number of lost packets $m \leq w_i(r) - 3$ with probability $\sum_{m=1}^{w_i(r)-3} \binom{m}{w_i(r)} l^m (1-l)^{w_i(r)-m}$ so that lost packets can be recovered by FR. In this case, $w_i(r)$ packets can be scheduled, the (r+1)-th round starts with probability 1, $T_i(r+1) = T_i(r) - (T_i^{w_i(r)} + \sum_{m=1}^{w_i(r)-3} \binom{m}{w_i(r)} l^m (1-l)^{w_i(r)-m} T_i^m)$, $w_i(r+1) = sst_i(r+1) = \lfloor w_i(r)/2 \rfloor$.
- Case 3-3: With packet loss, but number of lost packets is large enough with probability $\sum_{m=w_i(r)-2}^{w_i(r)} \binom{m}{w_i(r)} l^m (1-l)^{w_i(r)-m}$ so that lost packets can not be recovered by FR. In this case, $w_i(r)$ packets can be scheduled, but there is no enough time for RTO recovery, thus the (r+1)-th round will not start, i.e., $T_i(r+1) = 0$ and $w_i(r+1) = 0$.

B. Case-4 Sub-Cases

- Case 4-1: Same as Case 3-1.
- Case 4-2: Same as Case 3-2.
- Case 4-3: With packet loss and number of lost packets ($m > w_i(r) - 3$) with probability $\sum_{m=w_i(r)-2}^{w_i(r)} \binom{m}{w_i(r)} l^m (1-l)^{w_i(r)-m}$ can be 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 $w_i(r+1) = w_{min}$.

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

VIII. Performance Evaluation

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

A. Simulation Settings

We used NS3 simulator with IEEE 802.15.4g and LTE communication protocols. The 802.15.4g bandwidth is set to 100 $kbps$ and LTE bandwidth is set to 10 $Mbps$. In the simulation, each data node delivers 100 packets to data center with 100 bytes of payload generated with $\lambda = 10/s$, $NP_{thr} = 3$, $1 \leq cwnd_{min} \leq 5$, $10 \leq cwnd_{max} \leq 20$, $sst_{start} = 16$, $20s \leq RTO\,Timer \leq 80s$, $K = \infty$ for data nodes and $rwnd = \infty$ for data center. In the simulation, channel busy probability p_b for each data node and path loss rate l are computed and applied in the path scheduling.

Various node deployment scenarios are simulated to show the performance of the proposed MPTCP techniques performs under different network configurations. These node deployment scenarios emulate IoT applications such as smart meter, smart agriculture and smart factory. We placed data center *DC* at the center and corner of node deployment area, respectively. The center placement is to show how the proposed MPTCP methods perform with shorter paths and less congested bottlenecks. On the other hand, the corner placement aims to demonstrate the the performance of the proposed MPTCP methods with longer paths and more congested bottlenecks.

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. We used conventional MPTCP (C-MPTCP) with Fastest-RTT scheduler and standard NewReno algorithm as baseline.

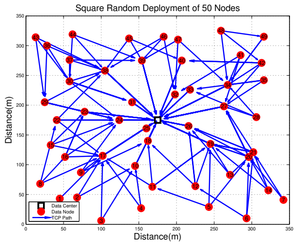

Fig. 2. Random Deployment of 50 Nodes with Node DC at Center

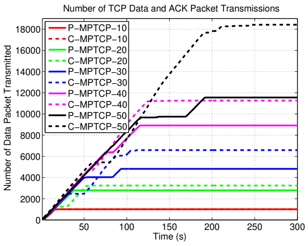

Fig. 3. Number of TCP Packet transmissions

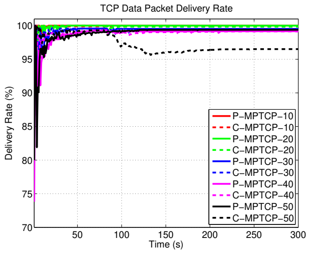

Fig. 4. TCP Data Packet Delivery Rate

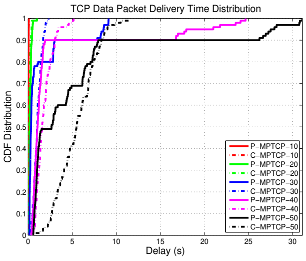

Fig. 5. TCP Data Packet Delay Distribution

B. Square and Rectangle Deployment of 10-50 Nodes with DC at Center

In these deployments, all nodes are 802.15.4g nodes. Data center is deployed at the center of deployment area and 9, 19, 29, 39 and 49 data nodes are randomly deployed in 150m×150m square, 200m×250m rectangle, 250m×300m rectangle, 300m×350m rectangle and 350m×350m square, respectively. Fig. 2 demonstrates the node deployment and MPTCP paths for 50 node case.

Fig. 3 shows the number of TCP data and ACK packet transmissions, where solid lines are transmissions by P-MPTCP and dash lines are transmissions by C-MPTCP. P-MPTCP reduces C-MPTCP packet transmissions from 1020 to 1011 by 1%, 3247 to 2279 by 30%, 6592 to 4824 by 27%, 11269 to 8915 by 21% and 18421 to 11554 by 37% for 10, 20, 30, 40 and 50 nodes, respectively. For both P-MPTCP and C-MPTCP, there are times where the variation of the packet transmission is not visible. This occurs when some data nodes finish their data delivery and the remaining data nodes are in RTO loss recovery, during which data nodes are waiting for timeout timer to be triggered and therefore, no new data packet is transmitted.

Fig. 4 demonstrates the TCP data packet delivery rate measured as the number of TCP data packets received by data center or data nodes divided by the number of TCP data packets transmitted or relayed. For 10-40 nodes, P-MPTCP and C-MPTCP achieve over 99% of delivery rate with P-MPTCP rate slightly higher. However, for 50 nodes, C-MPTCP delivery rate is 96.5% and P-MPTCP delivery rate is 98.3%, a 1.8% of improvement.

Fig. 5 shows the TCP data packet latency measured as the time difference between the time data packet is transmitted by source data node and the time data packet is received by data center. For 10-20 nodes, P-MPTCP and C-MPTCP have similar CDF delay distribution. For 30-50 nodes, P-MPTCP delivers 90% data packets faster than C-MPTCP does. C-MPTCP takes 41s, 81s, 132s, 160s and 266s to deliver 900, 1900, 2900, 3900 and 4900 data packets, respectively. P-MPTCP shortens times to 37s by 8%, 58s by 28%, 114s by 14%, 136s by 15% and 212s by 20%, respectively.

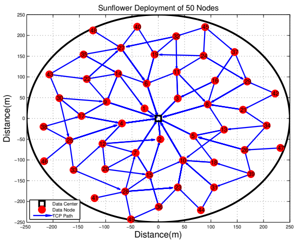

Fig. 6. Sunflower Deployment of 50 Nodes with Node C at Center

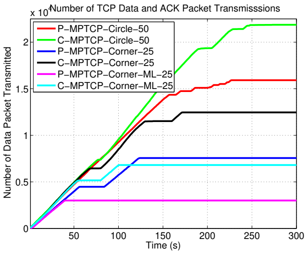

Fig. 7. Number of TCP Packet Transmissions

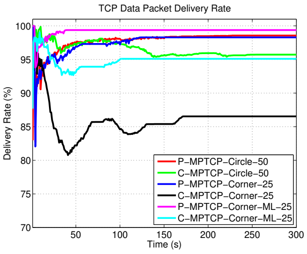

Fig. 8. TCP Data Packet Delivery Rate

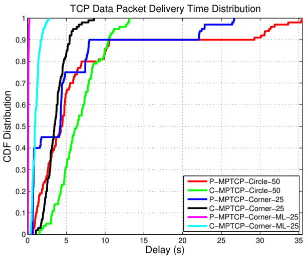

Fig. 9. TCP Data Packet Delay Distribution

In terms of data throughput, P-MPTCP improves C-MPTCP throughput from 17.5 kbps to 19.5 kbps by 2%, 18.8 kbps to 26.2 kbps by 7.4%, 17.6 kbps to 20.4 kbps by 16%, 19.5 kbps to 23.6 kbps by 21% and 14.7 kbps to 18.5 kbps by 26% for 10, 20, 30, 40 and 50 nodes, respectively.

C. Sunflower Deployment of 50 Nodes with DC at Center

In this simulation, 50 nodes are deployed in a circle of 250m radius using Sunflower deployment algorithm with data center at the center. This deployment is to show the impact of node density. All nodes are 802.15.4g nodes. Fig. 6 shows node deployment and MPTCP paths.

The Circle-50 curves in Fig. 7 shows the number of TCP packet transmissions. C-MPTCP transmits 21862 packets and P-MPTCP transmits 15915 packets, a 27% of transmission reduction. Compared to the square center deployment of 50 nodes, more packets are transmitted in this deployment due to the longer paths caused by the sparser node deployment. This result reveals that the longer path has more impact than the higher interference.

The Circle-50 curves in Fig. 8 shows the TCP data packet delivery rate. C-MPTCP delivery rate is 95.7% and P-MPTCP delivery rate is 98.6%, a 2.9% of delivery rate improvement.

The Circle-50 curves in Fig. 9 illustrates the TCP data packet latency. P-MPTCP delivers 80% data packets faster than C-MPTCP does. C-MPTCP takes 268s to deliver 4900

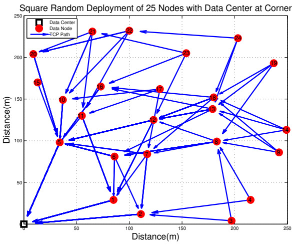

Fig. 10. Square Random Deployment of 25 Nodes without Multi-Link Node

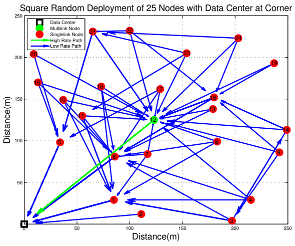

Fig. 11. Square Random Deployment of 25 Nodes with One Multi-Link Node

data packets. However, P-MPTCP takes 246s, an 8% of latency reduction.

For the TCP data throughput, P-MPTCP improves C-MPTCP throughput from 14.6 kbps to 15.9 kbps by 34%.

D. Square Deployment of 25 Nodes with DC at Corner

This deployment is to evaluate performance of P-MPTCP with longer paths and more congested bottlenecks and most importantly, to demonstrate the impact of ML node. Data center is placed at the corner and 24 data nodes are randomly deployed in a 250m×250m square. Figs. 10 and 11 show the network topology without and with ML node, respectively. Fig. 11 reveals that neighboring 802.15.4g nodes discover ML node 12 and build MPTCP paths through ML node 12.

The Corner-25 curves and Corner-ML-25 curves in Fig. 7 shows the number of TCP packet transmissions without ML node and with ML node, respectively. Without ML node, C-MPTCP transmits 12457 packets and P-MPTCP reduces transmissions to 7561 packets, a 39% of transmission reduction. With one ML node, C-MPTCP and P-MPTCP reduce their transmissions to 6807 packets and 3003 packets, a 45% and a 60% of transmission reduction, respectively. These results emphasize the impact of ML node for both C-MPTCP and P-MPTCP. With one ML node, P-MPTCP reduces 56% of C-MPTCP transmissions. In addition, 25 nodes take more transmissions to deliver 2400 packets than 30 nodes with data center at center to delivery 2900 packets. It reveals the impact of node deployment.

The Corner-25 curves and Corner-ML-25 curves in Fig. 8 demonstrates the TCP data packet delivery rate without ML node and with ML node, respectively. P-MPTCP achieves much higher data packet delivery rate. Without ML node, C-MPTCP data delivery rate is 86.5% and P-MPTCP data delivery rate is 98.3%, a 11.8% of improvement. With one ML node, C-MPTCP improves its delivery rate from 86.5% to 95.1% and P-MPTCP improves its delivery rate from 98.3% to 99.4%. P-MPTCP improves C-MPTCP data delivery rate by 4.3%.

The Corner-25 curves and Corner-ML-25 curves in Fig. 9 shows the TCP data packet latency without ML node and with ML node, respectively. P-MPTCP delivers data packets much faster than C-MPTCP Does. Without ML node, C-MPTCP takes 192s to deliver 2400 data packets, P-MPTCP takes 144s, a 24% of latency reduction. With one ML node, C-MPTCP takes 131s, P-MPTCP only takes 81s, a 38% of latency reduction. P-MPTCP significantly reduces packet latency and delivers 100% of packet faster than C-MPTCP does. These results reveal the efficiency of P-MPTCP and the impact of ML node.

Without ML node, P-MPTCP improves C-MPTCP TCP data throughput from 10 kbps to 14.7 kbps by 33%. With one ML node, P-MPTCP improves C-MPTCP TCP data throughput from 14.7 kbps to 23.7 kbps by 61%.

IX. Conclusion

The consumer IoT devices are evolving from current generation to next generation. Next generation IoT devices can support multiple communications interfaces and perform more functions. Accordingly, multipath networking technologies can be applied to improve IoT network performance. Multipath TCP (MPTCP) has achieved success in computer networks. However, its deployment over wireless IoT networks has not been well studied, especially for IoT networks using CSMA based communications protocols. This paper models IEEE 802.15.4 Non-Slotted CSMA algorithm as a Markov chain model and proposes MPTCP techniques, a path establishment method, an adaptive NewReno algorithm and a path scheduling mechanism with an innovative RTT computation method, for heterogeneous IoT networks to deliver data over multiple paths to data centers. Compared with conventional MPTCP using standard Fastest-RTT scheduler and NewReno congestion control algorithm, the proposed MPTCP techniques can reduce up to 56% packet transmission, shorten up to 38% data packet latency, improve up to 61% data throughput and increase up to 11.8% data delivery rate. In addition, multi-link nodes can significantly improve network performance, node density and node deployment can also impact network performance.

References

[1] “Multipath Tcp Linux Kernel Implementation,” http://multipath-tcp.org/pmwiki.php/Users/ConfigureMPTCP, 2018.

[2] V. Hapanchak and A. Costa, “A Cross-layer Approach for MPTCP Path Management in Heterogeneous Vehicular Networks,” *Journal of Communications Software and Systems*, vol. 19, 2023.

[3] 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.

[4] 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.

[5] 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.

[6] Fu, S. and Atiquzzaman, A., “Performance Modeling of SCTP Multihoming,” in *IEEE Global Communications Conference (GLOBECOM)*, 2005.

[7] C. Buratti, “Performance Analysis of IEEE 802.15.4 Beacon-Enabled Mode,” *IEEE Transactions On Vehicular Technology*, vol. 59, 2010.

[8] Y. H. Zhu, L. Jia, and Y. Zhang, “Enhancing Channel Contention Efficiency in IEEE 802.15.4 Wireless Networks,” *Sensors*, 2022.

[9] C. Buratti and R. Verdone, “A Mathematical Model for Performance Analysis of IEEE 802.15.4 Non-Beacon Enabled Mode,” in *14th European Wireless Conference*, 2008.

[10] K. Govindan, A. Azad, K. Bynam, and S. Patil, “Modeling and Analysis of Non Beacon Mode for Low-Rate WPAN,” in *12th Annual IEEE Consumer Communications and Networking Conference (CCNC))*, 2015.