



Throughput stability and flow fairness enhancement of TCP traffic in multi-hop wireless networks

M. Joseph Auxilius Jude¹ · V. C. Diniesh¹ · M. Shivanranjani²

Published online: 21 May 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

In the multi-hop wireless network, transmission control protocol (TCP) throughput stability and flow fairness performances are worsened due to slower flow convergence in the loss recovery phase and flat-rate reduction during the congestion control process. In this article, a combination of network-assisted and window utilization based congestion control approach, known as feedback assisted improved recovery + (FAIR+), is proposed to overcome TCP's limitations under multi-hop wireless networks. The proposed FAIR+ algorithm initiates the congestion control process based on the queue level notification of the relay node and trims down the sending rate based on TCP flow's utilization level. In the congestion recovery phase, the FAIR+ algorithm implements a newer window increment pattern that achieves a faster convergence rate than the existing RFC 6582 implementation. The throughput stability of the FAIR+ algorithm is validated using the duality model and multi-hop wireless simulation. The simulation results convincingly prove that the FAIR+ attains significant improvement in throughput, flow fairness, and end-to-end latency performance over the existing TCP variants (OQS, NRT, and Westwood).

Keywords Multi-hop wireless network (MHWN) · Transmission control protocol (TCP) · Explicit congestion notification (ECN) · Active queue mechanism (AQM) · Additive increase and multiplicative decrease (AIMD)

1 Introduction

Recent years witnessed an escalation in the deployment of wireless multi-hopping technologies in accessing the internet via 4G/5G cellular links [1–3] and Wi-Fi hotspots [4–7]. The multi-hop communications considerably widen the network coverage and boost the system capacity of the

cellular and Wi-Fi networks. The multi-hop device-to-device (D2D) communication remains the vibrant research topic in 5G networks [8–10] as it improves spectrum efficiency and attains better fairness in sharing the network resource. The collection of self-governing and self-healing wireless nodes combined to form multi-hop wireless networks (MHWN) [11–14]. Wireless nodes in MHWN share information among themselves through a chain of intermediate wireless relay nodes without any pre-existing communication infrastructure. The wireless relay nodes function as a router that can maintain, update, repair, and reconstruct routes autonomously. The MHWN, with its ubiquitous computing capability, self-configuration, and faster deployment, makes them an excellent choice for emergency response communication, battlefield communication, disaster communication, and community wireless network. Figure 1 shows the schematic representation of multi-hopping at the last-mile cellular wireless network.

In wired or wireless internet, TCP [15] remains a dominant transport protocol as it handles an enormous volume of global internet data traffic and reliably delivers

✉ M. Joseph Auxilius Jude
jude2193@gmail.com

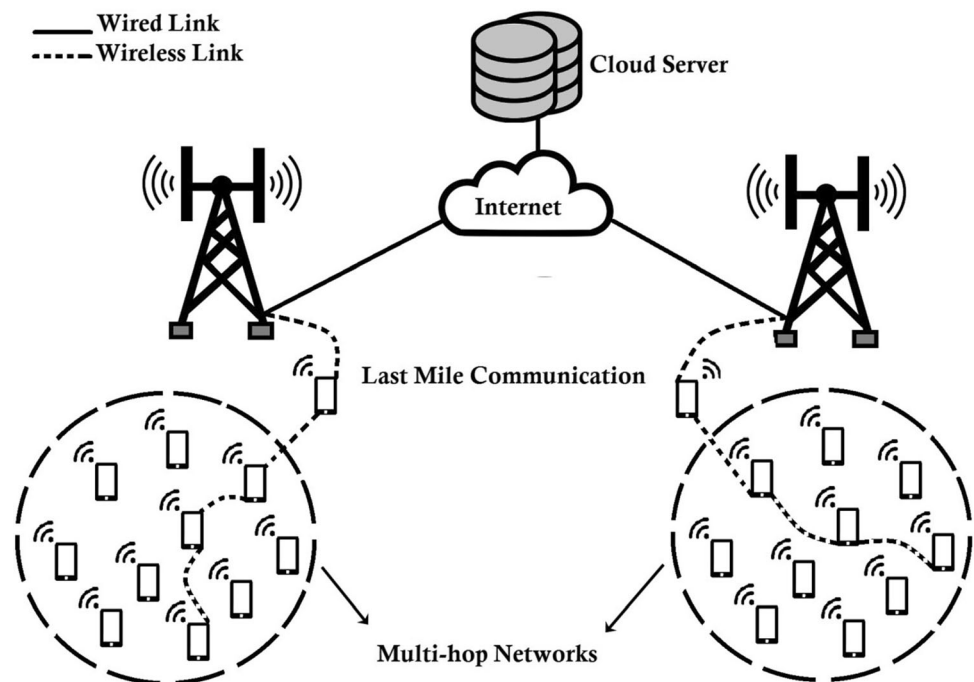
V. C. Diniesh
vcdiniesh@gmail.com

M. Shivanranjani
ranjani093@gmail.com

¹ Vinton Network Lab, Department of Electronics and Communication Engineering, Self Organised Networking Group (SONG), Kongu Engineering College, Perundurai, Tamil Nadu 638060, India

² Department of Electronics and Communication Engineering, MPNMJ Engineering College, Chennimalai, Tamil Nadu 638112, India

Fig. 1 Multi-hop networks deployed at last mile cellular wireless network



data packets between the end devices. TCP plays an influential role in delivering data traffic between wireless end systems and the remote cloud server in the 5G enabled internet of things (IoT) for smart city applications [16–19]. In recent years, an extensive array of experiments and studies were carried out to validate the performance of TCP under the different channel and traffic conditions of ultra-dense 5G networks [20–23]. TCP’s congestion control algorithm remains the prime research agenda due to its responsibility in maintaining the system throughput rate below the network capacity, i.e., controlling the packet transmission rate to avoid buffer overflows at the wireless relay nodes. For each connection, TCP maintains two state variables congestion window ($cwnd$) at the source and the receiver window ($rwnd$) at the destination that regulates the data flow into the network based on network capacity or receiver buffer capacity. TCP’s congestion control process is governed primarily by the slow start (SS) phase, congestion avoidance (CA) phase, and loss or congestion recovery phase.

The sender fixes the threshold limit for the SS phase, commonly known as SS_{Thresh} based on the receiver’s advertised capacity during the connection negotiation phase. In the SS phase, the $cwnd$ doubles for each acknowledgment (ACK) packet, and the sender swiftly invokes the CA phase as the $cwnd$ growth exceeds the SS_{Thresh} . In the CA phase, $cwnd$ growth is incremented one packet for each round trip time (RTT) until the missing ACK or packet loss is detected. On experiencing the packet loss through missing ACK or timeout, TCP invokes the

congestion control process by flatly reducing the transmission rate by half of its current $cwnd$ size. In the congestion recovery phase, the sender gradual increment $cwnd$ size by one packet for each RTT until the attainment of steady-state condition ($cwnd = \text{network capacity}$).

The congestion control algorithms under MHWN are categorized broadly into independent and network-assisted congestion control approaches. The independent congestion control approaches remain vulnerable against RTT delay spike impact, which occurs due to the rerouting of data packets in the wireless relay nodes associated with the node mobility. Several research studies and experimental investigations in the past [24–28] reveals that the throughput stability of TCP traffic deteriorates under the wireless environment due to spurious initiation of a congestion control process for non-congested packet drops (link failure and channel errors). However, the network-assisted congestion control approach remains the preferred choice in wireless conditions that minimize the impact of the spurious rate reduction based on the explicit congestion notification (ECN) [29, 30] from the intermediate nodes. The sender responds to the ECN message by initiating the congestion control process, thereby maintaining flow stability and preventing network congestion at its infancy stage.

In the MHWN, the three significant limitations that cripple throughput, fairness, and end-to-end message latency performances of the standard TCP implementation are

- (i) TCP, with its weakly designed congestion control algorithm, fails to recognize non-congested packet losses (channel error and link failure losses) and initiates a spurious congestion control process, results in severe throughput degradation.
- (ii) The flat-rate reduction initiated by the sender during the congestion control process severely affects the bandwidth sharing capability of TCP connection with the lowest utility rate.
- (iii) In the loss recovery phase, the *cwnd* growth with a slower flow convergence rate requires more RTTs to reach the steady-state condition that result in more extended time duration to complete the data flow between the source–destination processes.

The FAIR+ algorithm proposed in this article suppresses traditional TCP's limitations by implementing an aggressive *cwnd* growth pattern, utilization level based rate reduction process, and faster window growth in the loss recovery process. The subsequent sections of this article are arranged as follows; Sect. 2 concisely discusses the background and related works of network-assisted congestion control approaches. Section 3 describes the TCP FAIR+ working mechanism. Section 4 derives the analytical model on throughput stability using duality theory. Section 5 outlines the simulation results and analysis of proposed TCP FAIR+ with the standard TCP variants and Section 6 with the conclusion.

2 Background and related works

This section briefly discusses various ECN based congestion control algorithms proposed for multi-hop wireless environment. The ECN implementation requires modification in the relay router queuing discipline and initiates congestion notification when the packet ingress rate exceeds the queue threshold level. Jersey [31] and New Jersey [32] algorithms are the first of its kind to introduce the ECN approach in the relay node to alert the end stations about network congestion. Furthermore, these algorithms rely on available bandwidth estimation (ABE) of Westwood [33] for fixing new SS_{Thresh} value after each window reduction process. However, these algorithms successfully regulate spurious rate reduction impact but fail to achieve convergence rate in the loss recovery phase due to slow *cwnd* growth pattern.

MAD-TCP [34] implementation requires modification in the medium access control (MAC) and internet protocol (IP) packet headers to warn the sender about buffer overflows at the relay routers, MAC layer contention, link failures, and channel errors. The sender initiates the rate reduction or route repair process based on the internet

control message protocol (ICMP) notifications. The feedback assisted congestion control method [35] requires feedback value computation based on ECN and the updated advertised window (*awnd*) value from the ACK packet to initiate the congestion control process. The link-layer originated explicit link status notification (LL-ELSN) [36] approach prevents the sender from spurious timeout during the packet losses associated with the link failure and channel errors based on the explicit re-transmission start notification of the relay node.

The joined TCP bandwidth allocation (TBA) algorithm and wireless explicit congestion notification (WECN) mechanism [37] estimates the optimal *cwnd* size that the destination can handle during each reception process, and the WECN mechanism intended to initiate an early warning to the sender about the congested wireless link. During ECN marking, the receiver process fixes a suitable *cwnd* size in the advertised window that the destination can handle without oppressing the wireless link capacity.

The non-congestion events (NCE) [38] implementation detects non-congested losses from the bottleneck link losses based on the ECN approach. The second approach differentiates the non-congested losses by comparing flight size information of the packets with link latency threshold value. However, the NCE algorithm fails to address the flat-rate reduction and slower convergence rate of the traditional NewReno approach. The non-congested re-transmission timeout (NRT) [39] comprises of NRT-detection, NRT-differentiation, and NRT-reaction. In NRT-detection, the sender verifies the ACK packets marked with the ECN flag as the implication of the bottleneck wireless link and cuts down the current transmission rate by half of its capacity. In NRT-differentiation and reaction, the sender interprets packet loss of unmarked ACK packets as the spurious timeout and initiate re-transmission without reducing the current transmission rate. NRT congestion control algorithm capable of curbing unnecessary rate reduction during non-congested losses neglects the limitations of flat-rate reduction and slower convergence rate.

The joint congestion control and scheduling (JCCS) [40] approach determine the minimum transmission rate of the sender based on the virtual throughput estimation approach. The congestion control process is initiated based on explicit feedback from the wireless relay node. However, the window-based approach results in a slower recovery rate during the loss or congestion recovery process. TCP optimal queue selection (OQS) [41] is a joint effort of sender-side TCP, and the wireless relay node's dual queue algorithm, intended to address the flow fairness and throughput stability issues in the asymmetric wireless network. In the dual queue approach, the primary buffer holds the data packets, and the secondary buffer holds the ACK packets. The primary and secondary buffer

implements unique queuing disciplines, i.e., active queue management (AQM) for primary and first-in, first-out (FIFO) for secondary buffer. At the sender side, the congestion control process is initiated based on the primary buffer ECN mechanism. During the congestion control process, the sender implements flat-rate reduction similar to that of the New Reno approach.

The feedback-assisted recovery (FAR) [42] implements concave window growth to achieve faster converge during the recovery process. The FAR uses three decisive points to fix the new SSThresh during the rate reduction process. However, FAR implementation requires more tuning parameters to achieve faster flow convergence compared to the traditional AIMD approach. Inter vehicular access network (IVAN) TCP [43] developed for vehicular environment follows the concave *cwnd* growth pattern and push more data packets into the network. However, during the node mobility and channel error conditions, IVAN experiences an enormous amount of packet drops compared to the linear window increment pattern.

In summary, the ECN based approaches yield a better throughput performance under a multi-hop wireless environment. However, two inherent limitations compromise the throughput and flow fairness performance of these approaches, (1) flat dispatching rate reduction during the congestion control process and (2) slow flow convergence in the loss recovery phase. The proposed FAIR+ approach is designed to overcome these limitations by implementing the GBRC algorithm during the rate reduction process and AWR algorithm in the loss recovery phase *cwnd* growth.

3 Proposed TCP FAIR+ congestion control approach

The proposed TCP FAIR+ implementation requires modifications in the SS phase *cwnd* growth pattern, congestion control process, and loss recovery algorithm.

3.1 SS phase window growth

In the SS phase, the sender gradually increments the transmission rate (*cwnd*) until it finds the network capacity. The proposed FAIR+ algorithm fixes the initial SS_{Thresh} as the maximum window size (*Max_wnd*) and partitions the entire *cwnd* growth into low and high utilization regions based on the growth threshold level δ , ($\delta = \frac{Max_wnd}{2}$). In the first RTT, the sender fixes *cwnd* value as initial window size [44] and subsequent RTTs; the sender doubles

the *cwnd* size for the successful reception of each ACK packet. In the region below $\frac{\delta}{8}$, *cwnd* is updated as

$$cwnd_{(k+1)} = cwnd_{(k)} + 1, \quad cwnd < \frac{\delta}{8} \tag{1}$$

When the *cwnd* growth rate exceeds $\frac{\delta}{8}$, the FAIR+ implements a newer increment pattern that allows the source node to grab the available network bandwidth quickly, and the *cwnd* is updated as

$$cwnd_{(k+1)} = cwnd_{(k)} + ((2 * Max_wnd) - cwnd_{(k)}) * \gamma \tag{2}$$

The parameter γ is the rate increment factor and takes the value $\frac{1}{4}$. The lesser γ value results in a slower growth rate, and the immense value leads to buffer overflows in the bottleneck routers. When the *cwnd* matches with *Max_wnd*, the sender swiftly initiates the CA phase and the *cwnd* is updated as

$$cwnd_{(k+1)} = cwnd_{(k)} + \frac{1}{cwnd_{(k)}} \tag{3}$$

The pseudocode for the SS and CA phase is presented as

Algorithm 1: SS and CA phase *cwnd* increment

```

cwnd → congestion window size
Max_wnd → maximum window
 $\delta \rightarrow \frac{Max\_wnd}{2}$ 
 $\gamma \rightarrow \frac{1}{4}$ 
For every RTT
    if (cwnd < Max_wnd)
        if (cwnd <  $\frac{\delta}{8}$ )
            cwnd(k+1) = cwnd(k) + 1
        else (cwnd ≥  $\frac{\delta}{8}$ )
            cwnd(k+1) = cwnd(k) + ((2 * Max_wnd) - cwnd(k)) *  $\gamma$ 
        end
        else (cwnd ≥ Max_wnd) (CA phase)
            cwnd(k+1) = cwnd(k) +  $\frac{1}{cwnd_{(k)}}$ 
        end
    end

```

The SS phase of the FAIR+ algorithm implements a faster *cwnd* growth pattern compared to that of the existing TCP implementation [45] that allows the sender to grab the available network bandwidth quickly (Fig. 2).

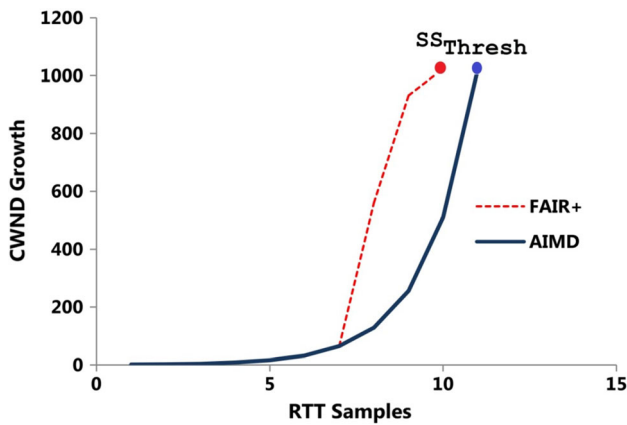


Fig. 2 SS phase cwnd growth of FAIR+ and AIMD

3.2 Congestion control process

The congestion in the relay node leads to performance degradation in the form of queuing latency, buffer overflows, and even collapse of the entire network. At the sender side, the congestion control process is initiated based on the two-level congestion notifications. The proposed dual queue dual marking (DQDM) (Fig. 3) algorithm has two significant functionalities.

- (i) The packet classifier drives the incoming data and control packets into appropriate queues (data (Q_D) and control (Q_C) buffers).
- (ii) The Q_C holds the non-data packets and ensures no loss of control packets in the relay routers. The Q_C implements a FIFO queuing mechanism. The Q_D accommodates the data packets and implements AQM [46, 47] that maintains different threshold levels to notify the queue build-up to the sender.

The control packets in the MHWN remain the lifeline for network survivability as it carries acknowledge packets and vital link related packets, such as route requests, route reply, link error, and link repair/update, message. The Q_C acts as a separate pipe to hold the very vital control

packets and occupies 10% of the total queue size (Q_T). For every packet arrival, Q_C compares the instantaneous queue size (Q_{Inst}) with the queue limits (Q_{limit}) and discards the packets when the Q_{Inst} exceeds Q_{limit} . The Q_D occupies 90% of Q_T and works on the following strategies (i) enqueueing data packets and (ii) congestion alert at regular intervals based on the queue accumulation levels (Queue threshold 1 (Q_{TH1}) and queue threshold 2 (Q_{TH2})).

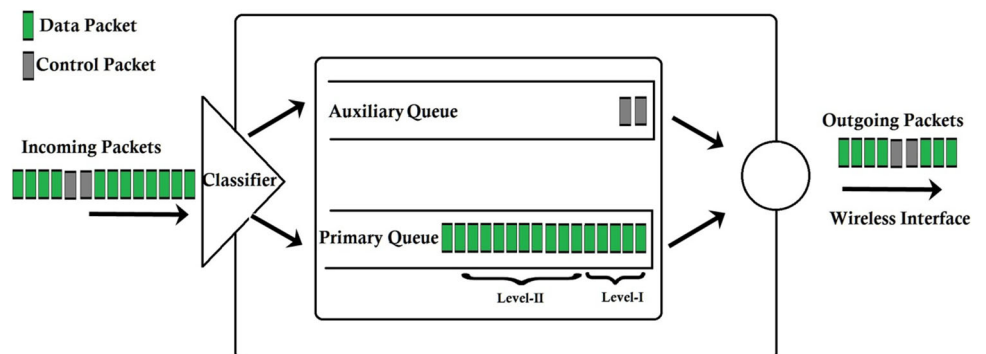
Fixing the queue threshold levels (Q_{TH1} and Q_{TH2}) remains a prime concern and the larger Q_{TH1} results in slower responsiveness and buffer bloat at the relay nodes. Conversely, a smaller Q_{TH1} value leads to frequent congestion notifications. For faster responsiveness towards queue build-up, the Q_D uses the minimum of 5 data packets as Q_{TH1} and a maximum of 15 data packets as Q_{TH2} . The ingenious idea behind two-level notifications is to alert the sender about the incipient or severe congestion in the wireless relay routers. The Q_D computes the average queue size (Q_{Avg}) [48] alternately of the actual queue size (Q_{cur}) with the highest marking probability ($M_p = 1$) and, the Q_{Avg} value is computed as

$$Q_{Avg}(K + 1) = Q_{Avg}(K) + Q_{wt}(Q_{cur}(K) - Q_{Avg}(K)) \tag{4}$$

where $Q_{Avg}(K + 1)$: Average queue size of ($K + 1$) packet, $Q_{Avg}(K)$: Average queue size of (K) packet, Q_{wt} : Queue Weight, $Q_{cur}(K)$: Current queue size of ($K + 1$) packet.

The relay router initiates congestion notifications (level 1 and level 2) when the packet arrival rate exceeds the first marking threshold Q_{TH1} and Q_{TH2} . The pseudo code for the packet processing in the Q_D is given as.

Fig. 3 The proposed DQDM model



Algorithm 2 *pseudo code for data queue (Q_D)*

```

Fixed Parameters
QTH1 = 5 packets of QD
QTH2 = 15 packets of QD
Initialization : QAvg(K+1) → 0
For each packet arrival Compute QAvg(K+1)
QAvg(K+1) = QAvg(K) + Qwi * ((Qcur(K) - QAvg(K))
if QAvg < QTH1
    Enqueue all the packet in QD
    else if QTH1 < QAvg < QTH2
        Mark CE-1 and Enqueue the packet
        else if QAvg > QTH2
            Mark CE-2 and Enqueue the packet
            else if QAvg > Qlimit
                Discard the incoming packet
            end
        end
    end
end
    
```

In the IP header, a one-bit congestion experience-1 (CE-1) and CE-2 field are added for two-level congestion notifications, as shown in Fig. 4a. For higher-layer support, an explicit congestion notification echo-1 (ECE-1) and ECE-2 flag are added in the TCP header, as shown in Fig. 4b. The relay router enables the CE-1 or CE-2 in the IP header for level 1 or level 2 notifications. Based on CE-1 or CE-2, the receiver sets the ECE-1 or ECE-2 flag in the ACK packets moving towards the sender. The receiver stops enabling the ECE-1 or ECE-2 echoes until the reception of the sender response flag, i.e., the congestion window reduced (CWR) flag. The overall communication model of the FAIR+ algorithm (Fig. 5) is developed based on the [49].

3.3 Rate adjustment and loss recovery

On receiving the ACK packet enabled with ECE-1, FAIR+ implements a growth based rate control (GBRC) algorithm that lessens the dispatching rate based on window utility threshold (δ). When the *cwnd* growth is in the low utilization region ($cwnd \leq \delta$), new *SS_{Thresh}* is computed as

$$SS_{Thresh} = \beta_1 * cwnd_{(k)}; \quad cwnd \leq \delta \tag{5}$$

Conversely, when the *cwnd* growth is in the high utilization region ($cwnd > \delta$), new *SS_{Thresh}* is computed as

$$SS_{Thresh} = \beta_2 * cwnd_{(k)}; \quad cwnd > \delta \tag{6}$$

For the second notification level, the sender fixes the new *SS_{Thresh}* irrespective of its utilization levels.

$$SS_{Thresh} = \beta_3 * cwnd_{(k)} \tag{7}$$

$$\beta_3 = 2 * (\beta_1 - \beta_2) \tag{8}$$

where β_1 ($\beta_1 = 0.7$) and β_2 ($\beta_2 = 0.5$) are the window reduction factors, larger (β_1 and β_2) value results in buffer overflows in the relay router and lesser value results in slower window recovery. In the congestion recovery phase, the proposed FAIR+ implements AWR algorithm which updates its *cwnd* based on new linear equation as follows

$$cwnd_{(k+1)} = cwnd_{(k)} + \frac{(\eta * Max_wnd)}{cwnd_{(k)}} \tag{9}$$

where η is the window growth factor and takes the value 2, larger η value results in faster oscillation, and the lesser η value results in slower window recovery. The pseudocode for the rate adjustment and loss recovery approach is given

Fig. 4 **a** IP header modification and **b** TCP flag modification

Version	Header Length	Type of Service(TOS)		Total Length			
0 1 0 0		DSCP	CE-1				
Identification			IP Flags		Fragment Offset		
			CE-2	D	M		

(a)

Header Length (4 bit)	Reserved (2 bit)	C	E	E	U	A	P	R	S	F
		W	C	C	R	C	S	S	Y	I
		R	E	E	G	K	H	T	N	N
			1	2						

(b)

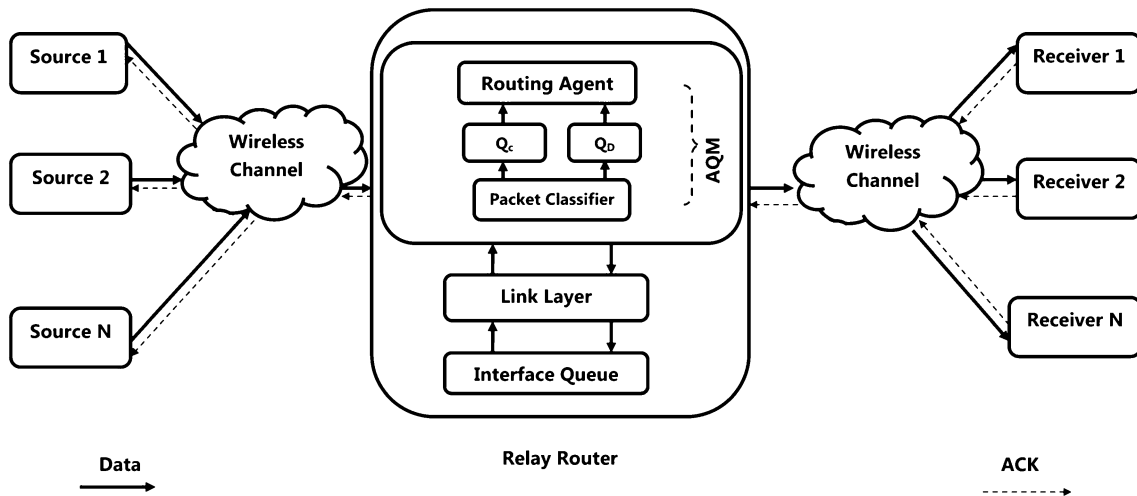


Fig. 5 FAIR+ communication model

as.

Algorithm 3 pseudo code for FAIR+ rate adjustment and loss recovery

```

β1, β2 and β3 = Increment factors
β1 = 0.7, β2 = 0.5 and β3 = 2*(β1 - β2)
For every RTT
  if ECE-2 received
    SSThresh = cwnd(k) * β3
    SSThresh = cwnd(k)
    cwnd(k+1) = cwnd(k) + (η * Max_wnd) / cwnd(k)
  else if ECE-1 received
    if cwndi > δ
      set SSThresh = cwnd(k) * β2
      SSThresh = cwnd(k)
      cwnd(k+1) = cwnd(k) + (η * Max_wnd) / cwnd(k)
    else (cwndi ≤ δ)
      set SSThresh = cwnd(k) * β1
      SSThresh = cwnd(k)
      cwnd(k+1) = cwnd(k) + (η * Max_wnd) / cwnd(k)
    end
  end
end
if (3DUPACK) / timeout
  retransmit the lost packet
end
end
    
```

The proposed FAIR+ algorithm performance is validated using two research methodologies (i) analytical model and (ii) simulation experiments. Finite state machine for FAIR+ algorithm is presented in Fig. 6

4 Throughput stability model

TCP throughput stability model based on duality theory [50] derives the best possible transmission rate x_s that the sender can achieve a maximum link utility $U_s(x_s)$ without exceeding the network capacity. Figure 7 represents closed-loop feedback model of network-assisted congestion control approach.

The throughput stability model of the FAIR+ algorithm considers source node transmission rate x_s as primal variable and wireless relay node’s queue accumulation level as dual variable p . The sender initiates rate reduction based on relay node queue length, commonly known as link price (or) congestion measure of the link. The maximum link utility $U_s(x_s)$ depends on the following factors, wireless link capacity (L_c), the number of senders (s), minimum transmission rate (mx_s), and the maximum transmission rate (Mx_s). The link capacity L_c depends on the signal to interference and noise ratio (SINR) of the radio channel, derived as [51]

$$L_c = W \log(1 + K\gamma) \tag{10}$$

where γ is the instantaneous interference of wireless link l , W is the signal bandwidth, and constant K is fixed depending on bit-error-rate, modulation, and the coding rate. To achieve the optimal transmission rate x_s , the sender needs to satisfy the following conditions, i.e., the primal variable x_s objective function is defined as

$$\max_{mx_s \leq x_s \leq Mx_s} \sum_s U_s(x_s) \tag{11}$$

$$\text{Subject to } \sum_s x_s \leq L_c, \forall l \tag{12}$$

The first condition states that the sender attains a maximum link utility $U_s(x_s)$ by choosing the transmission rate

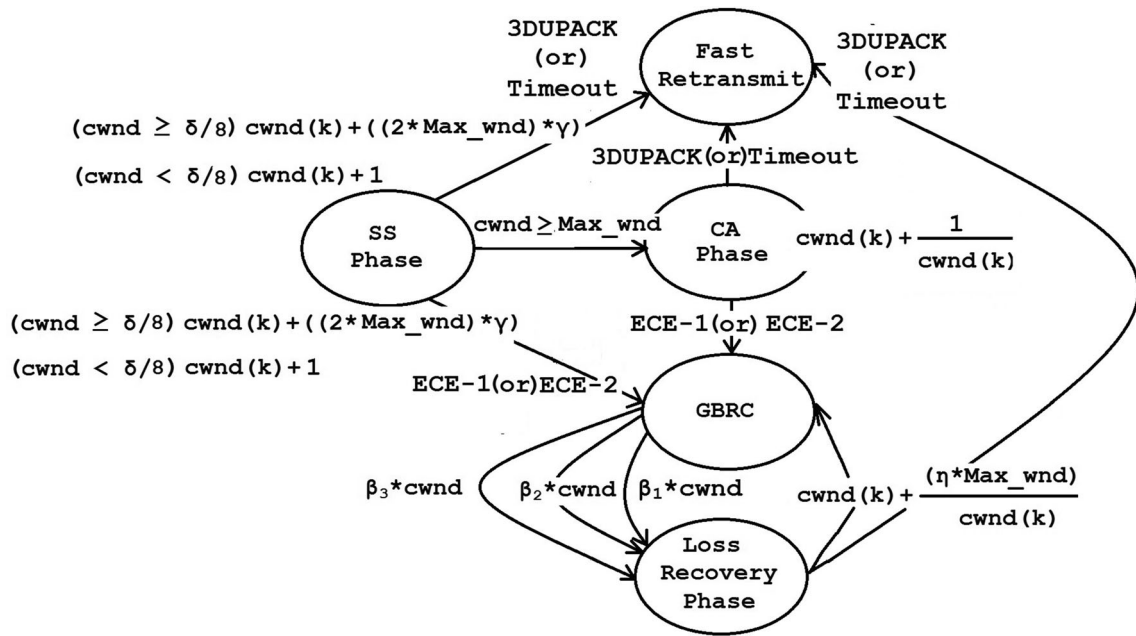
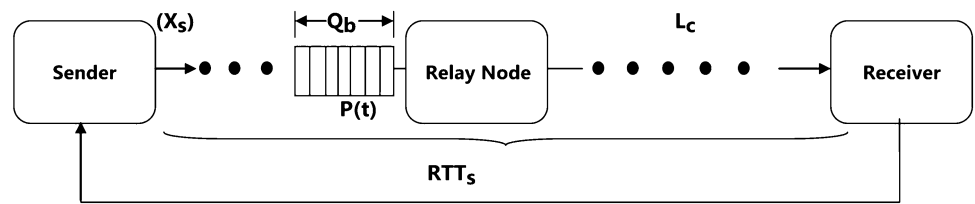


Fig. 6 FAIR+ state transition diagram

Fig. 7 TCP closed loop feedback model



x_s between mx_s and Mx_s , where $mx_s \geq 0$ and $Mx_s \leq \infty$. The second condition states that the aggregate optimal transmission rate x_s does not exceed link capacity L_c . Condition 1 and 2 are attained by deriving the duality Eq. (13).

$$\min_{p \geq 0} D(p) = \sum_s \left(\max_{mx_s \leq x_s \leq Mx_s} U_s(x_s) - x_s(p) \right) + \sum_L p * L_c \tag{13}$$

where $U_s(x_s)$ is a strictly concave function or non-decreasing function, i.e., higher utility results in a larger transmission rate. The $x_s(p)$ is the inverse of the utility function $(U_s(p)^{-1})$ and exist between mx_s , and Mx_s .

$$x_s(p) = \left[U_s(p)^{-1} \right]_{mx_s}^{Mx_s} \tag{14}$$

The transmission rate x_s is the function $(f(x_s = p))$ of the congestion measure variable p . In the equilibrium (or) steady-state condition, the source node transmission rate x_s equals the link capacity. The sender infers equilibrium state through congestion measure variable p (queue length information). In the equilibrium condition, utility function $U_s(x_s)$ remains within p

$$U_s(x_s) = p; \quad \forall s \tag{15}$$

For each sender, $U_s(x_s)$ is the function of x_s ($x_s = p_s$)

$$U_s(x_s) = \int f_s(x_s) dx, \quad x_s \geq 0 \tag{16}$$

The second term in Eq. (13) denotes the capacity per link and congestion price per link.

The proposed FAIR+ algorithm uses the relay node’s queue length information as a congestion measure variable $(p(t))$ and reduces transmission rate based on TCP flows current $cwnd$ growth level. Let the equilibrium window size be w_s , and RTT_s be the round trip time delay. The average data transmitted per unit time $x_s(t)$ is given by

$$x_s(t) = \frac{w_s(t)}{RTT_s} \tag{17}$$

The sender increments window size for every successful acknowledgment with probability $(1-d)$ and reduces the transmission rate based on loss probability, 'd', ($d = p$). The FAIR+ algorithm implements two different rate reduction approaches based on the w_c growth level. Let r_1 be the rate reduction factor for $cwnd$ with lesser growth rate and r_2 be the rate reduction factor for $cwnd$ with the higher

growth rate. From Eqs. (5) and (6), the reduced transmission rate for high (r_1) and low (r_2) rate flows are computed as

$$r_1 = \frac{7}{10}; \quad cwnd \leq \delta$$

$$r_1 = \frac{1}{2}; \quad cwnd > \delta$$
(18)

The flow with the highest window growth (r_2) is taken into consideration for the analysis as the goal of the optimization algorithm is to find the maximum link utility $U_s(x_s)$. Furthermore, the window increment rate of the proposed AWR algorithm in the loss recovery phase varies between $1/w_s$ and $4/w_s$ (Eq. (9)). For every positive acknowledgment ($1-d$), the window is updated per unit time as

$$\frac{4x_s(t)(1-d(t))}{w_s(t)}$$
(19)

Similarly, for ECN marked acknowledgment (d), the rate is reduced by $w_s/2$ and the window is decremented per unit time as

$$\frac{x_s(t)w_s(t)d(t)}{2}$$
(20)

At equilibrium, average window increment is balanced by average window decrement per unit time

$$\frac{4x_s(t)(1-d(t))}{w_s(t)} = \frac{x_s(t)w_s(t)d(t)}{2}$$
(21)

The loss rate $d(t)$ is derived by substituting Eq. (17) in Eq. (21)

$$d(t) = \frac{8}{8 + x_s^2(t)RTT_s^2}$$
(22)

We know that $x_s = p_s = d$, substituting Eq. (22) in Eq. (16)

$$U_s(x_s) = \int \frac{8}{8 + x_s^2(t)RTT_s^2} dx, \quad x_s \geq 0$$
(23)

After integrating Eq. (22), the maximum link utility $U_s(x_s)$ is derived as

$$U_s(x_s) = \frac{\sqrt{8}}{RTT_s} \tan^{-1} \sqrt{8x_s}RTT_s$$
(24)

The FAIR+ iterative algorithm can be derived by incorporating the DQDM congestion marking mechanism. Let Q_b be the queue size of the bottleneck wireless relay router, and Q_{Thresh} be the marking threshold. When the arrival rate λ exceeds the Q_{Thresh} , the wireless relay router initiates congestion notification with the highest probability ($m_p(t) = 1$), as shown in Fig. 8.

As illustrated in Fig. 8, the marking probability in $m_p(t)$ of DQDM is derived as

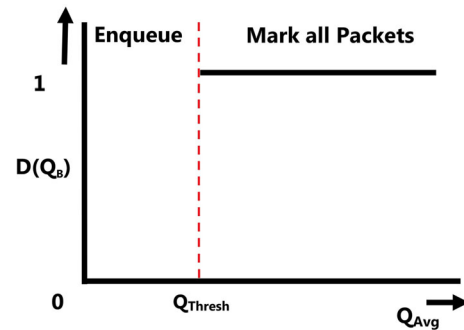


Fig. 8 DQDM packet marking probability

$$m_p(t) = 0; \quad Q_b < Q_{Thresh}$$

$$1; \quad Q_b \geq Q_{Thresh}$$
(25)

The relationship between $w_s(t)$, ECN marked acknowledgment packet $d(t)$ at the link l and end-to-end ECN probability $m_p(t)$ per unit time is given by

$$m_p(t) = w_s(t)d(t)$$
(26)

In each RTT, sender increase the rate by $4/RTT_s$ with probability $1 - m_p(t)$ (positive acknowledgment) per unit time and trims down the window growth $w_s/(2RTT_s)$ with $m_p(t)$ probability. Now average change in window growth in time period t is given by

$$\frac{4(1 - m_p(t))}{RTT_s} - \frac{w_s(t)m_p(t)}{2RTT_s}$$
(27)

By substituting Eqs. (17) and (26) in Eq. (27)

$$\frac{4(1 - x_s(t)RTT_s d(t))}{RTT_s} - \frac{x_s^2(t)RTT_s d(t)}{2}$$
(28)

Equation (17) can be rewritten per unit time period ($1/RTT_s$) as

$$\frac{4(1 - x_s(t)RTT_s d(t))}{RTT_s^2} - \frac{x_s^2(t)d(t)}{2}$$
(29)

Based on the marking probability in $m_p(t)$, the rate adjustment mechanism of the proposed FAIR+ algorithm is derived as

$$x_s(t+1) = \left[x_s(t) + \frac{4(1 - x_s(t)RTT_s d(t))}{RTT_s^2} - \frac{d(t)x_s^2(t)}{2} \right]$$
(30)

Based on the Eqs. (22) and (30), the source rate of FAIR+ is compared with ECN enabled NewReno (Fig. 9) for 50 RTT samples, The following parameters are used for computation; $Q_{Thresh} = 5$ packets ($Q_{Thresh1} = 5$ packets and $Q_{Thresh2} = 15$ packets), RTT_s varies between 60 and 150 ms, $w_s(t)$ attains the maximum value of 1030 packets. The proposed FAIR+ attains an improved transfer rate than the standard NewReno approach under varying RTT

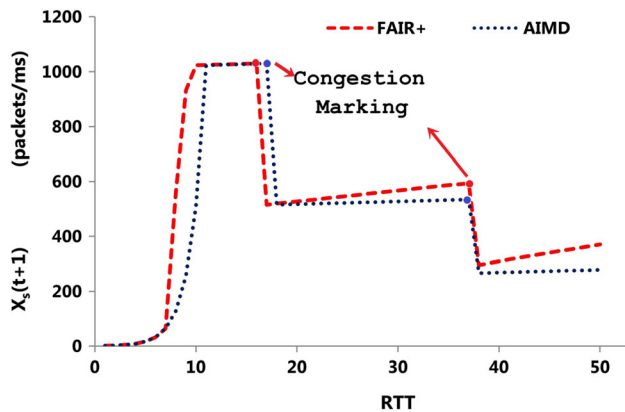


Fig. 9 Source rate of FAIR+ and NewReno/ECN

conditions. In the subsequent section, the duality model is validated by MHWN simulations under three diverse scenarios

5 Simulation and analysis

Multi-hop wireless simulation experiments carried out to validate the analytical model claim. The propagation and traffic models are created using network simulator 2 (NS-2) [52], and wireless node movements are generated using the Bonnmotion mobility generator [53]. A total of 204 simulated experiments were performed with a 95% confidence interval under the three patterns (hop length, traffic connections, and mobility) of the multi-hop wireless scenarios. As bounded by RFC 5166 [54], average throughput, average goodput, the average end-to-end latency, and flow fairness metrics were chosen for the analysis.

- (i) Average throughput is the measure of successful data packets received over the radio communication channel during the specified time interval.
- (ii) Average goodput is the measure of successful reception of actual (excluding header) data/packet/information rate over the radio communication channel during the specified time interval.
- (iii) Average end-to-end data packet latency is the measure of RTT delay between the sending and receiving process.
- (iv) Flow fairness is the measure of resource sharing among the different RTT competing flows.

Table 1 gives precise details on simulation parameters used for the MHWN simulation.

5.1 Hop length analysis

The path or hop length analysis measures the FAIR+ algorithm and existing approaches performance under multi-

Table 1 Simulation parameters for indoor and outdoor wireless models

Simulation parameters	
No. of nodes	500
Simulation time	1000 s
Simulation area	1000 × 1000 m
Antenna type	Omni directional
Antenna height	1.5 m
Propagation model	Shadowing
Reference distance (d_0)	10 m
Path loss exponent (β)	2
Shadowing deviation σ (dB)	5
Carrier sense threshold range	120 m
Receiver threshold range	90 m
Frequency	2.4 GHz
System loss	1
Queue SIZE	1000 packets
Routing protocol	AODV
Mobility model	Random way point
Mobility	5, 10, 15 and 20 m/s
Traffic connections	2,4,6,8
TCP variants	FAIR+ , Westwood, OQS NRT
Window size	256
Packet size	1400 Bytes

hop length conditions. The multi-hop simulation experiments are carried out by constructing the chain topology with a minimum path length of 2 hops to a maximum of 10 hops. The performances of hop-length experiment analysis in Fig. 10 are summarized as follows; the FAIR+ algorithm under minimum and maximum hop length conditions attain an average throughput of 5920 KB/s and 1950 KB/s for the simulation duration of 1000 s. As the hop length increases, there is a considerable reduction in the transmission rate due to processing and queuing delays at intermediate wireless relay routers. However, under both (minimum and maximum) path length conditions, the FAIR+ algorithm attains an average throughput improvement of 31.58% and 34.35% against Westwood, 23.31%, and 17.43% improvement against OQS and 25.16% and 24.61% improvement against NRT approach. Similarly, the FAIR+ attains an average goodput of 5710 KB/s and 1780 KB/s over the existing congestion control approaches under both path length conditions. The proposed FAIR+ attains improvement in average goodput rate of 28.02% and 39.95% against Westwood, 23.11%, and 16.11% improvement in goodput against OQS, and

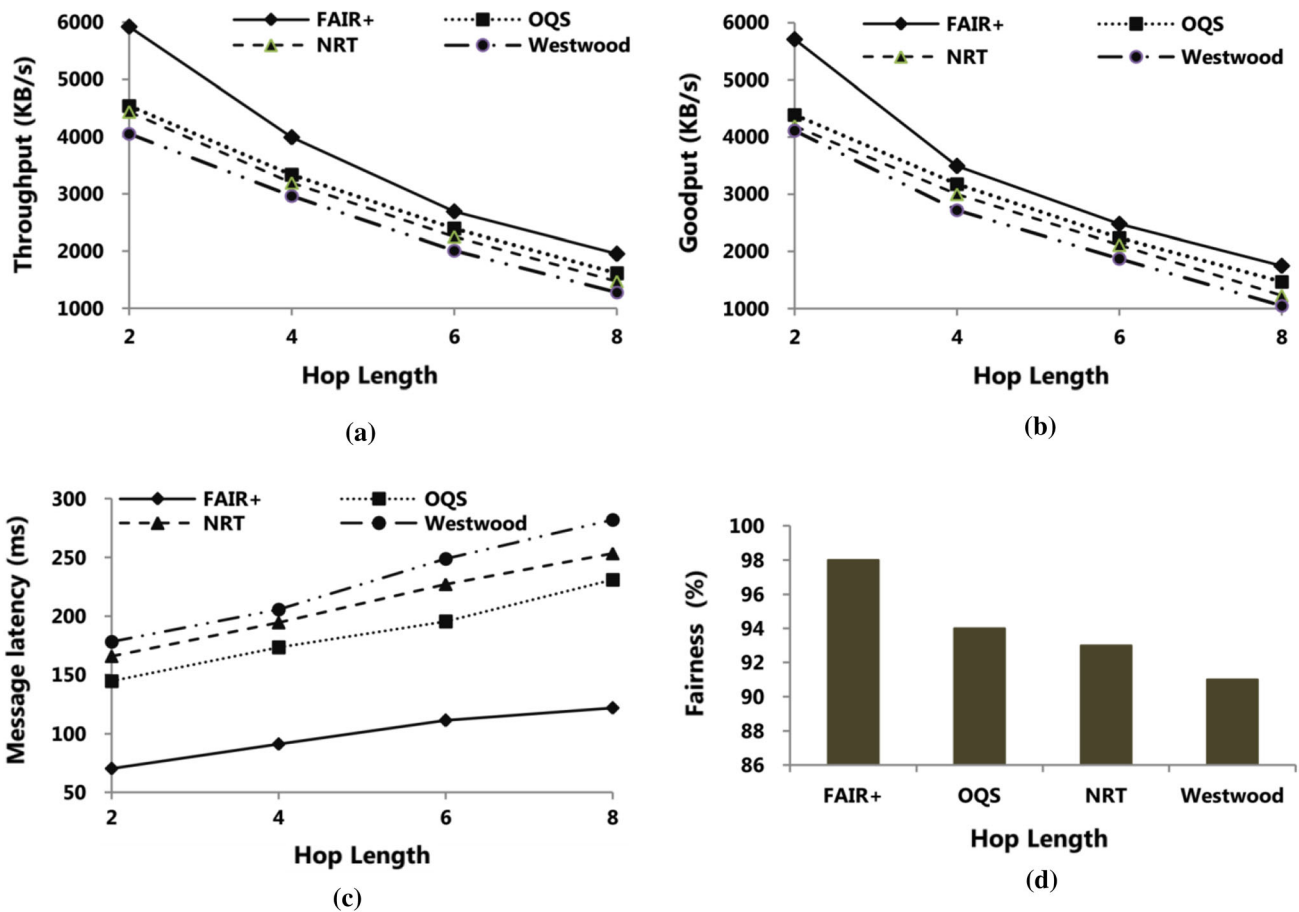


Fig. 10 Hop length versus Metrics **a** Average throughput (KB/s), **b** Average goodput (KB/s), **c** Average message latency (ms) and **d** Fairness (%)

26.61%, and 29.71% improvement in goodput against NRT under both hop length conditions.

The flow stability in the network path influences end-to-end message latency between sender-receiver processes. The DQDM algorithm implemented at the relay router responds quickly to the increase in the queue accumulation level by initiating ECN, and the sender maintains flow stability by regulating outbound network traffic that results in minimum end-to-end average message latency. For maximum and minimum path length conditions, the FAIR+ algorithm attains the substantial reduction in average message latency against (60.68%, and 56.94%) Westwood, (51.39%, and 47.64%) OQS, and (57.58%, and 52.14%) NRT approaches. TCP traffic connections with FAIR+ implementation attains the higher bandwidth sharing capability due to *cwnd* utilization based rate control during the ECN marking and AWR algorithm in the loss recovery phase. The FAIR+ attains the flow fairness of 97.8% against Westwood (91%), OQS (94%), and NRT (93%) approaches under path length conditions.

5.2 Traffic connections analysis

The traffic handling capacity of the proposed FAIR+ algorithm is measured by conducting simulation experiments under increasing traffic flows of two connections to a maximum of ten connections. The grid topology (22 × 22) was chosen for conducting traffic flow experiments and the traffic analysis performances are summarized as follows; the FAIR+ algorithm attains a throughput of 4670 KB/s and 1530 KB/s for light load and heavy load conditions (Fig. 11). In the increasing traffic flow conditions, the FAIR+ algorithm initiates rate reduction during incipient congestion based on the relay node’s queue accumulation level, thereby maintaining flow stability with improved throughput and goodput rate over the existing approaches. Under light and heavy load conditions, the FAIR+ algorithm yields an average throughput improvement of 36.65% and 35.35% against Westwood, 31.11%, and 14.37% against OQS and 35.33% and 24.18% against NRT approach. Similarly, the FAIR+ attains a better goodput of 4410 KB/s and 1410 KB/s for light and heavy traffic conditions. Under both traffic conditions, FAIR+ attains a

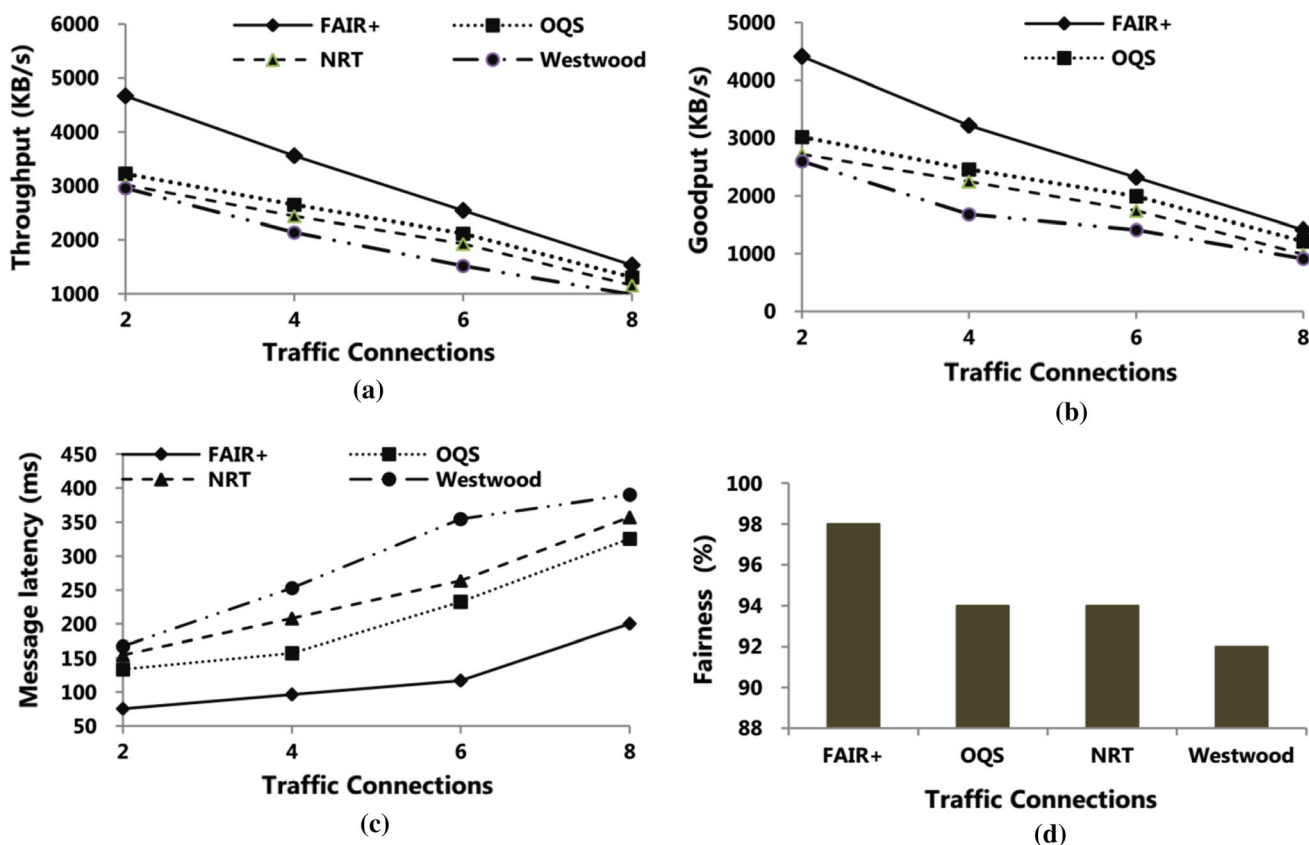


Fig. 11 Traffic flows versus Metrics a Average throughput (KB/s), b Average goodput (KB/s), c Average message latency (ms) and d Fairness (%)

considerable improvement in the goodput rate over its counterparts, i.e., goodput improvement of 35.33% and 24.18% against Westwood, 35.33%, and 24.18% against OQS, and 35.33% and 24.18% against NRT approach.

The average end-to-end message latency performance in the high traffic conditions are influenced by regulating the arrival rate within the relay router’s processing capability. The FAIR+ rate adjustment based on queue length notification achieves better flow stability in the relay routers. In high traffic conditions, FAIR+ attains 48.71%, 38.58%, and 43.96% reduction in average message latency against Westwood, OQS, and NRT. Similarly, FAIR+ attains a 98% flow fairness due to its *cwnd* utilization based reduction and AWR algorithm during the loss recovery phase. Conversely, the existing approaches attain the flow fairness of 92% (Westwood), 94% (OQS), and 94% (NRT) due to its flat-rate reduction approach during the congestion control process and *cwnd* growth in the loss recovery phase.

5.3 Mobility analysis

The node mobility experiments were conducted to measure the FAIR+ congestion control algorithm performance

under different node mobility conditions ranging from 5 m/s to a maximum of 20 m/s. The following observations were noted from the node mobility analysis. In the wireless environment, frequent link breakage and route association takes place between the mobile nodes due to the fast movement of wireless nodes, results in packet drops. The FAIR algorithm with ECN markings identifies the non-congested packet loss and prevents needless transmission rate reduction, which substantially improves its throughput and goodput performance. The FAIR+ algorithm attains an average throughput of 3730 KB/s and 940 KB/s for low and high node mobility conditions (Fig. 12). FAIR+ algorithm attains throughput improvement of 18.76% and 39.89% against Westwood, 7.51%, and 13.82% against OQS and 15.54% and 26.7% against the NRT approach under low and high node mobility conditions.

The FAIR+ attains the average goodput rate of 3591 KB/s and 880 KB/s for high and low mobility conditions, i.e., 18.11% and 42.95% improvement against Westwood, 8.63% and 16.93% improvement in OQS, and 16.21% and 32.72% improvement against NRT. Furthermore, FAIR+ attains the substantial reduction in average end-to-end message latency performance due to its queue accumulation assisted rate control that maintains the TCP

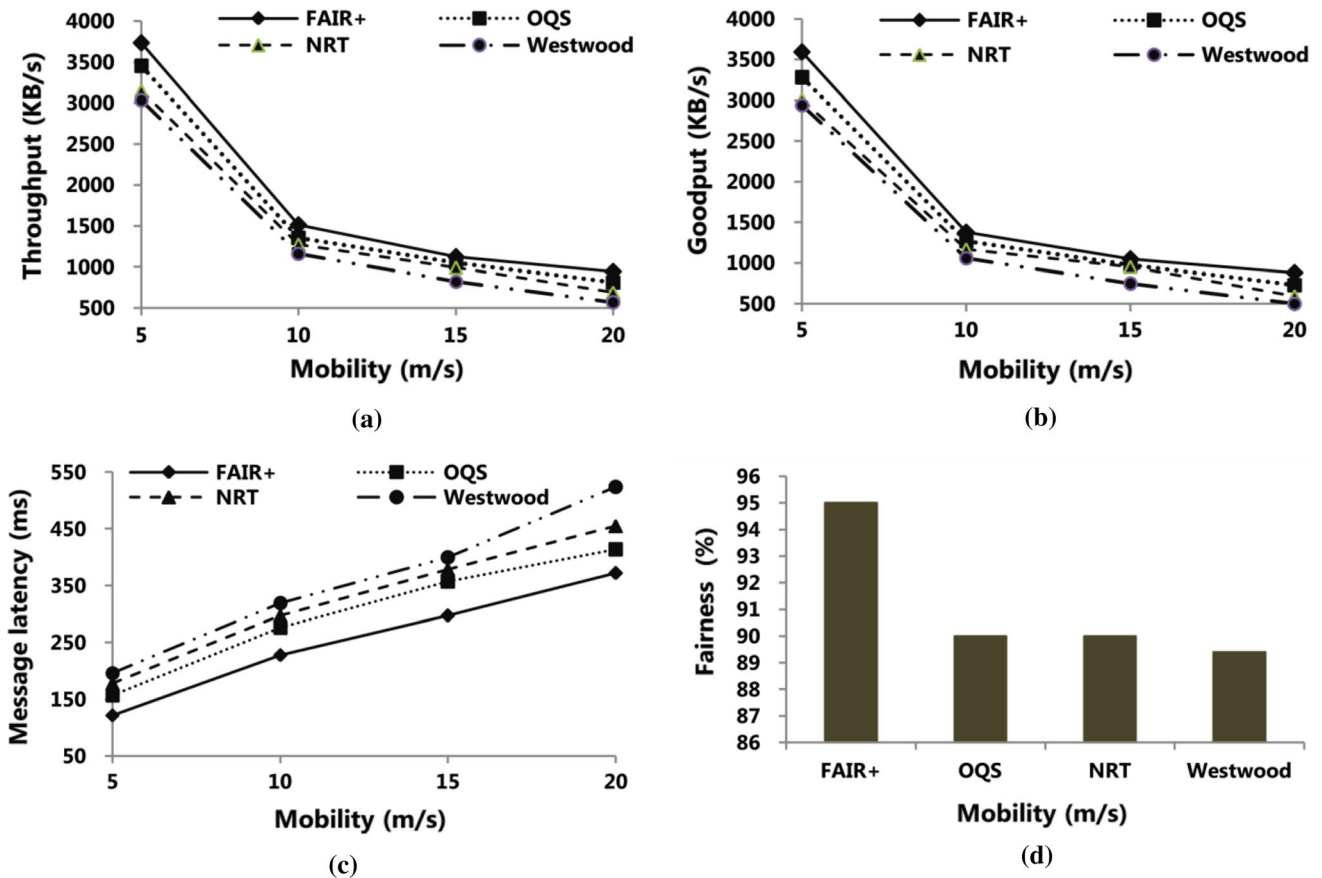


Fig. 12 Node mobility versus Metrics **a** Average throughput (KB/s), **b** Average goodput (KB/s), **c** Average message latency (ms) and **d** Fairness (%)

traffic within the network capacity. Under both mobility conditions, the FAIR+ algorithm yields a lesser average latency of 38.17% and 29.12% against Westwood, 22.51%, and 10.12% against OQS, and 31.72% and 18.25% against NRT approach. In node mobility conditions, the FAIR+ algorithm attains a 95% flow fairness performance against Westwood (89.4%), OQS (90%), and NRT (90%) approaches.

6 Conclusion

The ECN based FAIR+ approach proposed in this article implements the DQDM, GBRC, and AWR algorithms to suppress the limitations of flat-rate reduction, and slower flow convergence under the multi-hop wireless environment. At the relay node, the DQDM algorithm initiates level 1 (incipient) and level 2 (severe) congestion notifications based on the queue accumulation level. At the sender side, the GBRC algorithm initiates rate reduction based on the *cwnd* utilization levels that considerably reduces flat-rate reduction limitations and allows the fair sharing of network resources among different TCP flows.

In the congestion recovery phase, the AWR algorithm implements a faster *cwnd* increment pattern, results in a faster steady-state convergence rate. The performance of the FAIR+ algorithm is analytically validated using the duality theory-based throughput stability model, which derives the best possible transmission rate based on window utility and queue length information. Similarly, simulation analysis proves that the FAIR+ attains a considerable improvement in average throughput, average goodput and flow fairness performances, and a significant decrease in average end-to-end message latency performance against other schemes under different wireless scenarios.

References

- Zhou, K., Gui, J., & Xiong, N. (2017). Improving cellular downlink throughput by multi-hop relay-assisted outband D2D communications. *EURASIP Journal on Wireless Communications and Networking*, 2017(1), 209. <https://doi.org/10.1186/s13638-017-0998-9>.

2. Shaikh, F. S., & Wismüller, R. (2018). Routing in multi-hop cellular device-to-device (D2D) networks: A survey. *IEEE Communications Surveys & Tutorials*, 20(4), 2622–2657. <https://doi.org/10.1109/COMST.2018.2848108>.
3. Venkataraman, H., Gandhi, D., & Tomar, V. (2014). Multi-hop multi-band intelligent relay-based architecture for LTE-advanced multi-hop wireless cellular networks. *Wireless Personal Communications*, 75(1), 131–153. <https://doi.org/10.1007/s11277-013-1352-0>.
4. El Alami, M., Benamar, N., Younis, M., & Shahin, A. A. (2019). A framework for enabling Internet access using Wi-Fi peer-to-peer links. *Wireless Personal Communications*, 109(1), 521–538. <https://doi.org/10.1007/s11277-019-06577-7>.
5. Fadlullah, Z. M., Kawamoto, Y., Nishiyama, H., Kato, N., Egashira, N., Yano, K., et al. (2017). Multi-hop wireless transmission in multi-band WLAN systems: Proposal and future perspective. *IEEE Wireless Communications*, 26(1), 108–113. <https://doi.org/10.1109/MWC.2017.1700148>.
6. Le, M., Clyde, S., & Kwon, Y. W. (2019). Enabling multi-hop remote method invocation in device-to-device networks. *Human-Centric Computing and Information Sciences*, 9(1), 1–22. <https://doi.org/10.1186/s13673-019-0182-9>.
7. Mellott, M. J., Garlisi, D., Colbourn, C. J., Syrotiuk, V. R., & Tinnirello, I. (2019). Realizing airtime allocations in multi-hop Wi-Fi networks: A stability and convergence study with testbed evaluation. *Computer Communications*, 145, 273–283. <https://doi.org/10.1016/j.comcom.2019.07.006>.
8. Huang, C., Zhai, B., Tang, A., & Wang, X. (2019). Virtual mesh networking for achieving multi-hop D2D communications in 5G networks. *Ad Hoc Networks*, 94, 101936. <https://doi.org/10.1016/j.adhoc.2019.101936>.
9. Mohamed, E. M., Elhalawany, B. M., Khallaf, H. S., Zareei, M., Zeb, A., & Abdelghany, M. A. (2020). Relay Probing for Millimeter Wave Multi-Hop D2D Networks. *IEEE Access*, 8, 30560–30574. <https://doi.org/10.1109/ACCESS.2020.2972614>.
10. Aniktar, H., & Bulus, U. (2016). Relay multi-hop communications for next generation mobile networks: Investigation of path-loss models. *Wireless Personal Communications*, 88(4), 897–910. <https://doi.org/10.1007/s11277-016-3218-8>.
11. Khanna, G., & Chaturvedi, S. K. (2018). A comprehensive survey on multi-hop wireless networks: milestones, changing trends and concomitant challenges. *Wireless Personal Communications*, 101(2), 677–722. <https://doi.org/10.1007/s11277-018-5711-8>.
12. Hoque, M. A., Hong, X., & Dixon, B. (2014). Efficient multi-hop connectivity analysis in urban vehicular networks. *Vehicular Communications*, 1(2), 78–90. <https://doi.org/10.1016/j.vehcom.2014.04.002>.
13. Ali, A., & Khan, F. A. (2020). Condition and location-aware channel switching scheme for multi-hop multi-band WLANs. *Computer Networks*, 168, 107048. <https://doi.org/10.1016/j.comnet.2019.107048>.
14. Coll-Perales, B., Gozálviz, J., & Friderikos, V. (2016). Context-aware opportunistic networking in multi-hop cellular networks. *Ad Hoc Networks*, 37, 418–434. <https://doi.org/10.1016/j.adhoc.2015.09.004>.
15. Allman, M., Paxson, V., & Blanton, E. (2009). RFC 5681: TCP Congestion Control. <https://tools.ietf.org/html/rfc5681>
16. Gomez, C., Arcia-Moret, A., & Crowcroft, J. (2018). TCP in the Internet of Things: From ostracism to prominence. *IEEE Internet Computing*, 22(1), 29–41. <https://doi.org/10.1109/MIC.2018.112102200>.
17. Pokhrel, S. R., & Williamson, C. (2018). Modeling compound TCP over WiFi for IoT. *IEEE/ACM Transactions on Networking*, 26(2), 864–878. <https://doi.org/10.1109/TNET.2018.2806352>.
18. Gotin, M., Lösch, F., & Reussner, R. (2019). TCP-Inspired Congestion Avoidance for Cloud-IoT Applications. In *IEEE international conference on software architecture companion (ICSA-C)*. (pp. 5–10). <https://doi.org/10.1109/ICSA-C.2019.00009>
19. Park, M., & Paek, J. (2019). TAIM: TCP assistant-in-the-middle for multihop low-power and lossy networks in IoT. *Journal of Communications and Networks*, 21(2), 192–199. <https://doi.org/10.1109/JCN.2019.000017>.
20. Zhang, M., Polese, M., Mezzavilla, M., Zhu, J., Rangan, S., Panwar, S., et al. (2019). Will TCP Work in mmWave 5G cellular networks? *IEEE Communications Magazine*, 57(1), 65–71. <https://doi.org/10.1109/MCOM.2018.1701370>.
21. Na, W., Lakew, D. S., Lee, J., & Cho, S. (2019). Congestion control vs. link failure: TCP behavior in mmWave connected vehicular networks. *Future Generation Computer Systems*, 101, 1213–1222. <https://doi.org/10.1016/j.future.2019.07.065>.
22. Le, H. D., Nguyen, C. T., Mai, V. V., & Pham, A. T. (2019). On the throughput performance of TCP cubic in millimeter-wave cellular networks. *IEEE Access*, 7, 178618–178630. <https://doi.org/10.1109/ACCESS.2019.2959134>.
23. Polese, M., Jana, R., & Zorzi, M. (2017). TCP and MP-TCP in 5G mmWave networks. *IEEE Internet Computing*, 21(5), 12–19. <https://doi.org/10.1109/MIC.2017.3481348>.
24. Al-Jubari, A. M., Othman, M., Ali, B. M., & Hamid, N. A. W. A. (2011). TCP performance in multi-hop wireless ad hoc networks: challenges and solution. *EURASIP Journal on Wireless Communications and Networking*, 2011(1), 198. <https://doi.org/10.1186/1687-1499-2011-198>.
25. Begin, T., Baynat, B., Lassous, I. G., & Abreu, T. (2016). Performance analysis of multi-hop flows in IEEE 802.11 networks: A flexible and accurate modeling framework. *Performance Evaluation*, 96, 12–32. <https://doi.org/10.1016/j.peva.2015.12.003>.
26. Rezaei, S., Gharib, M., & Movaghar, A. (2018). Throughput analysis of IEEE 802.11 multi-hop wireless networks with routing consideration: A general framework. *IEEE Transactions on Communications*, 66(11), 5430–5443. <https://doi.org/10.1109/tcomm.2018.2848905>.
27. Maity, M., Raman, B., & Vutukuru, M. (2017). TCP Download performance in dense WiFi scenarios: Analysis and solution. *IEEE Transactions on Mobile Computing*, 16(1), 213–227. <https://doi.org/10.1109/tmc.2016.2540632>.
28. SreeArthi, D., Malini, S., Jude, M. J. A., & Diniesh, V. C. (2017). Micro level analysis of TCP congestion control algorithm in multi-hop wireless networks. *IEEE International Conference on Computer Communication and Informatics (ICCCI)*. <https://doi.org/10.1109/iccci.2017.8117765>.
29. Ramakrishnan, K., Floyd, S., & Black, D. (2001). RFC3168: The addition of explicit congestion notification (ECN) to IP. <https://tools.ietf.org/html/rfc3168>
30. Black, D. (2018). RFC8311: Relaxing Restrictions on Explicit Congestion Notification (ECN) Experimentation. <https://tools.ietf.org/html/rfc8311>
31. Xu, K., Tian, Y., & Ansari, N. (2004). TCP-Jersey for wireless IP communications. *IEEE Journal on Selected Areas in Communications*, 22(4), 747–756. <https://doi.org/10.1109/JSAC.2004.825989>.
32. Xu, K., Tian, Y., & Ansari, N. (2005). Improving TCP performance in integrated wireless communications networks. *Computer Networks*, 47(2), 219–237. <https://doi.org/10.1016/j.comnet.2004.07.006>.
33. Casetti, C., Gerla, M., Mascolo, S., Sanadidi, M. Y., & Wang, R. (2002). TCP Westwood: end-to-end congestion control for wired/wireless networks. *Wireless Networks*, 8(5), 467–479. <https://doi.org/10.1023/A:1016590112381>.
34. Chang, H. P., Kan, H. W., & Ho, M. H. (2012). Adaptive TCP congestion control and routing schemes using cross-layer

- information for mobile ad hoc networks. *Computer Communications*, 35(4), 454–474. <https://doi.org/10.1016/j.comcom.2011.11.008>.
35. Byun, H. J., & Lim, J. T. (2005). Explicit window adaptation algorithm over TCP wireless networks. *IEE Proceedings-Communications*, 152(5), 691–696. <https://doi.org/10.1049/ip-com:20045091>.
 36. Yun, J. H. (2009). Cross-layer explicit link status notification to improve TCP performance in wireless networks. *EURASIP Journal on Wireless Communications and Networking*, 2009(1), 617818. <https://doi.org/10.1155/2009/617818>.
 37. Peng, F., & Leung, V. C. (2007). Enhancing fairness and throughput of TCP in heterogeneous wireless access networks. *International Journal of Wireless Information Networks*, 14(3), 225. <https://doi.org/10.1007/s10776-007-0061-6>.
 38. Sreekumari, P., & Chung, S. H. (2011). TCP NCE: A unified solution for non-congestion events to improve the performance of TCP over wireless networks. *EURASIP Journal on Wireless Communications and Networking*, 2011(1), 23. <https://doi.org/10.1186/1687-1499-2011-23>.
 39. Sreekumari, P., & Lee, M. (2013). TCP NRT: A new TCP algorithm for differentiating non-congestion retransmission timeouts over multihop wireless networks. *EURASIP Journal on Wireless Communications and Networking*, 2013(1), 172. <https://doi.org/10.1186/1687-1499-2013-172>.
 40. Huang, P. K., Lin, X., & Wang, C. C. (2013). A low-complexity congestion control and scheduling algorithm for multihop wireless networks with order-optimal per-flow delay. *IEEE/ACM transactions on networking*, 21(2), 495–508. <https://doi.org/10.1109/TNET.2012.2213343>.
 41. Priya, S. S., & Murugan, K. (2015). Enhancing TCP fairness in wireless networks using dual queue approach with optimal queue selection. *Wireless Personal Communications*, 83(2), 1359–1372. <https://doi.org/10.1007/s11277-015-2455-6>.
 42. Jude, M. J. A., & Kuppuswami, S. (2017). Enhanced window increment and adaptive recovery TCP for multi-hop wireless networks. *Electronics Letters*, 53(6), 438–440. <https://doi.org/10.1049/el.2016.2601>.
 43. Jude, M. J. A., Diniesh, V. C., Shivarajani, M., & Shanju, R. (2018). A feedback aware reliable transport protocol with improved window increment mechanism for inter vehicular wireless network. *Wireless Personal Communications*, 98(1), 1119–1134. <https://doi.org/10.1007/s11277-017-4911-y>.
 44. Chu, J., Dukkupati, N., Cheng, Y., & Mathis, M. (2013). RFC6928: Increasing TCP's initial window. <https://tools.ietf.org/html/rfc6928>
 45. Henderson, T., Floyd, S., Gurtov, A., & Nishida, Y. (2012). RFC6582: The NewReno modification to TCP's fast recovery algorithm. <https://tools.ietf.org/html/rfc6582>.
 46. Adams, R. (2012). Active queue management: A survey. *IEEE Communications Surveys & Tutorials*, 15(3), 1425–1476. <https://doi.org/10.1109/SURV.2012.082212.00018>.
 47. Al-Saadi, R., Armitage, G., But, J., & Branch, P. (2019). A survey of delay-based and hybrid TCP congestion control algorithms. *IEEE Communications Surveys & Tutorials*, 21(4), 3609–3638. <https://doi.org/10.1109/COMST.2019.2904994>.
 48. Floyd, S., & Jacobson, V. (1993). Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 4, 397–413. <https://doi.org/10.1109/90.251892>.
 49. Singhal, C., & De, S. (Eds.). (2017). *Resource allocation in next-generation broadband wireless access networks*. Hershey: IGI Global.
 50. Low, S. H. (2003). A duality model of TCP and queue management algorithms. *IEEE/ACM Transactions on Networking*, 11(4), 525–536. <https://doi.org/10.1109/TNET.2003.815297>.
 51. Tran, N. H., Hong, C. S., & Lee, S. (2012). Cross-layer design of congestion control and power control in fast-fading wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 24(2), 260–274. <https://doi.org/10.1109/TPDS.2012.118>.
 52. The Network Simulator-2 (2020). <https://www.isi.edu/nsnam/ns>. Accessed April 26, 2020.
 53. BonnMotion (2020) A mobility scenario generation and analysis tool. <https://sys.cs.uos.de/bonnmotion>. Accessed April 26, 2020.
 54. Floyd, S. (2008). RFC 5166: Metrics for the Evaluation of Congestion Control Mechanisms. <https://tools.ietf.org/html/rfc5166>.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



M. Joseph Auxilius Jude received his B.E. in Electronics and Communication Engineering, M.E. with specialization in Digital Communication and Networking and Ph.D. under Anna University, Chennai, India in 2005, 2007 and 2019 respectively. He is with Self Organized Networking Group (SONG) of the ECE department at Kongu Engineering College. He is closely associated with the development of advanced wireless networking lab and organized

funded industry-aligned training programmes in alliance with Defence Research and Development Organisation (DRDO), Department of Science and Technology (DST) and Industry Institute Partnership Cell. He is a peer reviewer for IEEE Access, IEEE Sensor Letters and Wiley International Journal of Communication Systems. His research interests include designing, testing and validation of scheduling algorithm for wireless sensor networks, congestion control algorithms (TCP and AQM) for the mobile ad-hoc network, vehicular ad-hoc network and cognitive radio network. He is a member of IEEE (Computer Society and Sensor Council), life member of IETE and institution chair of IEEE computer society.



V. C. Diniesh received his B.E. in Electronics and Communication Engineering and M.E. with specialization in Embedded and Real Time System under Anna University, Chennai, India in 2009 and 2011 respectively. He is currently pursuing Ph.D. in the area of wireless ad-hoc and sensor networks. He is with Self Organized Networking Group (SONG) of the ECE department at Kongu Engineering College. He is closely connected with the routing protocol development

and training for tactical wireless networks especially for disaster and battlefield management, organized funded programmes in collaboration with DST and Industry Institute Partnership. His research interests include designing, testing and validation of routing protocols for mobile ad-hoc network and wireless sensor network. He is a member of IEEE and life member of IETE.



M. Shivaranjani completed her Master Degree in Communication Systems, from Kongu Engineering College affiliated to Anna University, Chennai. She received her Bachelor Degree in Electronics and Communication Engineering from Kongu Engineering College affiliated to Anna University, Chennai. She is with Self Organized Networking Group (SONG) and currently working as an assistant professor in MPNMJ Engineering College.

Her current research focuses on designing Transmission Control

Protocol (TCP) for Wireless Multi-hop Networks especially vehicular ad hoc network, Mobile ad hoc network and Cognitive Radio Network.