

# A Combined TCP-friendly Rate control with WFQ Approach for Congestion Control for MANET

**Y. Narasimha Reddy**

Royalaseema University, Kurnool, A. P., India  
E-mail: [narasimhareddy.phd@gmail.com](mailto:narasimhareddy.phd@gmail.com)

**P V S. Srinivas**

Sreenidhi Institute of Technology and Science, Hyderabad, India  
E-mail: [pvssrinivas@sreenidhi.edu.in](mailto:pvssrinivas@sreenidhi.edu.in)

Received: 20 January 2018; Accepted: 18 April 2018; Published: 08 June 2018

**Abstract**—Congestion control techniques are extensively used to avoid congestion over the wireless network. But these techniques are incapability of to handle the increased utilization of the various application which raising high congestion and packet loss over the network and causing inconvenient to different services. The TCP-friendly rate control (TFRC) protocol is primarily considered to describe the effective and finest potential provision for such applications which is following it preeminent in the wired and wireless environment. But it also suffers due to slow start and time-consuming process which required several round-trip-time (RTT) to reach an optimal level of the communication rate. As the TRFC transmission rate is highly affected by the increase RTTs and this results in an raise in the packet loss and a corresponding significant decrease in the throughput. In this paper, we propose an integrated TFRC with weighted fair queue (WFQ) approach to overcoming the congestion and minimize the RTTs. The WFQ mechanism manages the incoming heavy traffic to ease the data rate control for smooth data flow to improve throughput. The simulation evaluation of the approach shows an improvisation in throughput with the low delay in different data flow conditions.

**Index Terms**—TCP-friendly rate control, Congestion control, WFQ, MANET.

## I. INTRODUCTION

With the rapid development of Wireless network, traffic congestion control has become one of the most important concerns in the existing network services and traffic types to accommodate increasingly dissimilar range [1]. Congestion control mechanisms that facilitate different types of Wireless traffic to meet certain types of quality of service (QoS) constraints are becoming increasingly important. Several systems in the network environment monitor for impending congestion before it

occurs. The QoS design is a fundamental feature of next-generation IP routers to enable differentiated delivery and to ensure delivery quality for various service traffic [5],[20],[27]. Although the TCP protocol handles "data-oriented applications" firmly, so UDP protocol mostly being utilized for the streaming relevance.

Data-driven applications are able to bear extensive delays and changing speeds without problems but require extremely stable services. For these applications, the TCP protocol is suitable. However, streaming applications, like voice and video, can tolerate some damage, but fluctuations in transmission speed should be least and smooth. The importance of quality of service (QoS) is comparable to the evolution of recent telecommunication networks, which is considered by a very large heterogeneity [3], [24]. All applications requiring a certain level of assurance from the network, especially real-time video applications, have the excellence of service (QoS) prerequisites [10].

The concept of a relatively slow response to "congestion control algorithms" erstwhile established to performs with applications, where comparative soft transmission rates are essential [3], [6], [7]. The main suggestion of "TCP Friendly Rate Control (TFRC)" [4] to utilize an "equation model of Reno" for throughput improvisation [12], and to calculate the communication rate. An algorithm that does not use the "self-clocking principle" execute in TCP can demonstrate tremendous reconcile moment. Specifically, many RTTs could possibly need to adjust the participation rate to the bandwidth accessible on the network [31]. In [2], [28], [30] proposed some adaptation techniques to allocate enhanced parameter amendment under diverse context settings. However, the parameter amendment approach in this technique is based on the hypothesis that there is a known grouping of optimal parameter settings to which the technique is to be adjusted. On the other hand, the optimality of preferences depends on environmental factors that cannot be universally true, so adjustment might not be beneficial [11].

In recent years, queuing rules have been established that can play an important responsibility in the "congestion control" of outsized, distributed in data networks. In a data network, the server communicate to a particular communication line among the nodes of the network have " $N$  source-destination pairs" and " $N$  incoming streams" that carry data over the lines as shown in Fig. 1.

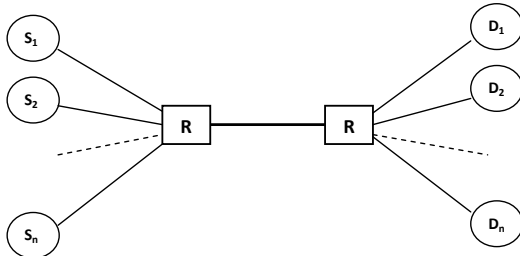


Fig.1. N Source-destination Pairs Transient Data above a Distinct Line

Maintaining "Fair Queue Management (FQM)" for every stream can control the stream and work well for streams that can saturate the stream or delay it to an unacceptable level. The fairness queuing mechanism fairly distributes the bandwidth of the line and provides a relatively small queue delay for small communication. The managing of an inaccessible communication line by itself does not assure proper managing of the network as a complete. On the other hand, it facilitates FQM which can be properly emulated and combined with reasonable buffer and rate control mechanisms to provide fair bandwidth allotment for multiple streams that can provide network-wide latency for communication and otherwise compromise in the network. This can provide reliability, high throughput, and overall network performance, especially in high traffic. In this paper, we propose an integrated approach of TFRC using Weighted Fair Queue (WFQ) [2],[19] to minimize "round-trip-time (RTT)" and "packet transmission loss" to accomplish high throughput and low latency.

The remaining paper is structured as follows. In the subsequently section-2, we reconsider various interrelated works. In Section-3, we discussed the proposed "TFRC with WFQ approach", Section-4, describes the simulation approach and results. Finally, in Section-5, we present the conclusion of the paper.

## II. RELATED WORKS

One of the key benefits of a wired network is its higher reliability and connection strength than its wireless counterparts. However, the performance of wired networks is often greatly degraded due to network congestion [21], [32]. Therefore we investigate other mechanisms for congestion control in the wired network to propose the most appropriate congestion avoidance mechanism needed.

### A. TCP in Congestion Control

At the present time, the wireless network is one of the

most significant communication systems in the world. The enormous of internet users accomplish a record high each year. New more people exploiting this standard to accumulate information or to make in touch with people throughout the world. Largely applications on the internet utilize "Transmission Control Protocol" (TCP) [6], [9]. TCP is a dependable connection-oriented protocol and has methods to recover from congestion. The TCP receiver transmits an acknowledgment to the sender to acknowledge receipt of the data. The transmission of the unacknowledged data packet is recurring until acknowledgment reaches the sender. TCP is believed to maintain the internet collectively, particularly due to its congestion control methods.

In most TCP congestion control mechanisms, if the TCP sender detects a missing data packet, it supposes that convenient congestion on the way to the destination network component. Therefore, the transmission speed is reduced very quickly to get used to congestion. If no further loss is detected, TCP will attempt to increase the transmission rate over again. Applications such as "multimedia streaming, online gaming, or voice-over-IP (VoIP)" earn additional revenue, even if end users have high-speed Internet access (such as DSL). These applications transmit constant data streams over long periods of time, unlike short-term connections for most "normal" TCP applications. The information that these applications send is important to the times. Consequently, it is not useful to use repeatedly because transmission of lost packets may be out of date when arriving delayed at the receiver. In addition, the application relies on a smooth data rate to transmit successive packet streams. Because TCP cannot satisfy these circumstances the "User Datagram Protocol" (UDP) is utilized as an alternative.

Because congestion is determined by traffic patterns and is not determined by the traffic routing mechanism, congestion cannot be permanently suppressed, but adverse impacts can be minimized by reducing network packet loss. The TCP supports mechanisms such as slow start, congestion prevention, fast retransmission and fast recovery, which is not very effective at reducing congestion despite reducing packet loss due to congestion [1], [8], [17], [18]. Therefore, we think that alternative congestion prevention mechanism is necessary. To avoid network congestion, researchers have advocated the use of an "active queue management" (AQM) strategy in which packets are dropped before the queue is full. Many AQM technologies have been reported, such as "adaptive virtual queues", "random early detection" (RED), random exponential marking, PI controllers, and blue and probabilistic blue schemes. Among these existing techniques, RED is actually one of the most widely used techniques [15],[17],[19].

### B. TFRC in Congestion Control

TFRC has been proposed to change the transmission rate more efficiently than TCP while sustaining TCP compatibility. Like TCP, TFRC also departs through a "slow start-up phase" immediately the following start-up,

increasing the transmission rate and reaching an equivalent distribution of bandwidth. The primary rate supported slow initiate practice is comparable to the TCP Windows supported slow commence the process. For TCP, the dispatcher approximately doubles the transmission rate for every RTT time. Even though the "slow-start algorithm", which twice the communication rate per calculated RTT, it may exceed the bottleneck of the linkage bandwidth.

TFRC is an "equation-based congestion control transport layer protocol" and is extremely unlike from the "window-based mechanism" of TCP. In "TFRC", the receiver estimates the failure incident rate and provides this information reverse to the transmitter. The "TFRC" transmitter afterward it computes and sets the new transmission rate using the throughput equation defined in RFC 5348 [4]. This equation makes available of the throughput that the TCP correlation receives beneath firmed-condition circumstances specified the defeat happening rate and the RTT of the association. With this procedure, "TFRC" is greatly enhanced at streaming audio and video streaming over time, with much lower throughput than TCP, and can compete moderately by means of TCP. Nevertheless, this causes "TFRC" to take action more slowly to modify the inaccessible bandwidth compared to TCP. In fact, the TFRC protocol performed experimental evaluations and simulations comparing TFRC and TCP Sack carried by Floyd et al.[23]. Their outcomes demonstrate superior fairness between challenging streams in "Drop-Tail", "RED queues" and low unpredictability in "TFRC throughput" evaluated to TCP.

TFRC approximates the diffusion rate supported on the "RTT session" to represent the "TCP congestion control" methods. It causes RTT injustice, which seriously influences the performance of elongated RTT streams [1], [8], [18]. Long RTT streams use less bandwidth than short RTT streams as soon as clients are provided flowing data on servers with diverse "end-to-end" transmission delays. Therefore, extensive RTT streams receive lower quality streaming data than short RTT flows. TFRC also calculates the communication rate supported by the packet failure happening rate slightly than the packet loss, reducing the fluctuation of the communication rate. Nevertheless, utilizing packet failure occurrence rates is less responsive whilst the network is jam-packed. This little down response causes a many packet loss while network congestion is rigorous.

### C. Fairness Queuing

Fair Queuing (FQ) is a new queue rule that includes applications that are critical to high-cost systems that prompted operations on data networks and services that support variable-size packets. "Queues and queuing algorithms" [2], [3], [19] are important components of traffic processing in the network to afford QoS. Queuing occurs simply when the interface is in use. If the interface is idle, the packet is sent without any special processing. Regular queues always use the FIFO principle. Packets with the longest latency are sent foremost. When the

queue is packed and extra packets are received, a tail drop occurs.

A further sophisticated queue mechanism typically uses multiple queues that classify the packets as a means by which the user can configure them and place them in the suitable queues. When the interface is prepared for transmission, the queue to which the subsequent packet is to be transmitted in particular according to the queuing algorithm. When the queue is ineffective, congestion takes place and the packet is discarded. Traffic overflow can be supervised by means of the appropriate queuing algorithm to classify the traffic and looking for a priority technique to forward the packet to the output link [22].

Queuing in networking comprise techniques such as "priority queuing", "fair queuing" (FQ) and "round robin queuing" [2], [3], [19], [26], [29]. FQ is meant to ensure that each flow has fair admittance to network resources and that the burst flow does not consume further a fair share of the output bandwidth. This scheme classifies packets into flows and is allocated to queues committed to that flow. Queues are served in a "round-robin fashion". FQ is as well referred to as flowing or "flow-based queuing" [3], [19]. The FQ mechanism consists of "Weighted Fair Queuing" (WFQ) [2], [25] and "Core-Stateless Fair Queuing" (CSFQ) [29]. For loss-sensitive applications, WFQ has proven to be the best fit because it has the least number of lost packets. It integrates TRFC and WFQ mechanisms to avoid seamless data flow and congestion.

## III. PROPOSED TFRC WITH WFQ APPROACH

The mechanism of the proposal initial presents an integrated model describing the model of data streaming. In second stage it describe the mechanism to control the TFRC flow control and Weighted Fairness Queue Control through WEQ Scheduler Scheme. Later, a mechanism of TFRC with WFQ based on Queue Flow Rate and Transmission Rate Estimation.

### A. Integrated Model

The massive streaming and multimedia applications traffic over the wireless is on the rise and we have proposed an integrated solution for TFRC that uses weighted fair queuing to provide solutions for TCP-friendliness and TFRC data rate control issues as shown in Fig. 2. These solutions are designed as either "router-based" or "end-to-end solutions" depending on where the explanation is utilized.

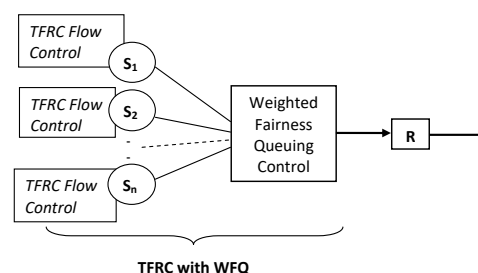


Fig.2. TFRC with WFQ integrated Model

The integrated model operates on two control mechanisms, "TFRC flow control" and "Weighted Fairness Queuing Control", to provide congestion and data rate control solutions. "Weighted Fair Queuing" (WFQ) was the first FQ implementation proposed, followed by several FQ implementations that attempted to simplify computational complexity or improve performance [2], [3]. We discuss each control mechanism and incorporate the functionality in the following sections.

### B. TFRC Flow Control Mechanism

The core of TFRC is then evaluated based on "packet loss event rate" as  $M$ , and "round-trip time" as  $RTT$ . To calculate approximately  $M$ , the term "lost event" consigns to multiple packets missing inside single  $RTT$ . The expression "loss interval" is explained as the number of packets among lost incidents.

The "TFRC" apply the complete standard failure period technique to estimate the standard failure happening rate throughout the preceding failure interval as  $v$ . Let  $P_t$  is described as "the number of packets of the  $t^{\text{th}}$  latest loss interval" and  $P_0$  is described as the intermission together with the packets arriving after the last failure. The overall standard failure period scheme acquires a "weighted average" of the preceding  $v$  periods and uses the same weight for mainly the recent  $v/2$  interval and a smaller weight for the previous interval. Thus, the average loss interval  $L_p$  is determined by a weight  $w_t$  as,

$$L_p = \sum_{t=1}^v \left( \frac{w_t P_t}{w_t} \right)$$

The weight  $w_t = 1$ , if  $1 \leq t \leq v/2$ , hence the packet loss event rate  $M$  will be calculated as  $1/L_p$ .

To estimate the  $RTT$ , the sender and receiver use the sequence number together. Each time the receiver sends the response, while this time the packet was received with the most recent data packet sequence number repeats. In this fashion, the transmitter determines the "RTT". The transmitter afterward utilizes the "exponentially weighted moving average" to smooth the determined  $RTT$  time. This "weight" establishes the sensitivity of the communication rate to transforms in "RTT time" [1].

Most rate control methods determine the transmission rate supported by network delay. Even though, sender with diverse "end-to-end transmission delays" characteristically accept feedback at altered rates. Therefore they will adjust the communication rate. This creates  $RTT$ -unfair problems. In order to resolve the "RTT-unfairness" difficulty, there have recently been recommending innovative "RTT-fair TCPs". In "TCP Hybla" it applies an invariable increasing algorithm [12], which offers "RTT-fairness" underneath assured constancy combinations.

The "TCP Vegas" will be providing superior "RTT-fairness", however, it ignores familiarity [15]. The "FAST TCP", enhances the permanence leaps of TCP Vegas, except when coexisted with the legacy TCP

protocol [26], the result is too passive or too aggressive. In addition, "Compound TCP" and "TCP Illinois", regulate  $RTT$ -imbalance with regulating the communication rate supporting on "queuing delay" [27], [28]. Because  $RTT$ -fairness is mostly affected by queue delay factors, we focus on minimizing delays and solving  $RTT$ -unfairness using the weighted fair queue control mechanism as described below.

### C. Weighted Fairness Queue Control Mechanism

Queuing is a mechanism that queues and removes queued packets from the buffer according to a quantity of scheduling method applied in the scheduler. Queues depend primarily above the quantity of the buffer and algorithm make use of to supervise the tailback. The "buffer amount" is an essential aspect of "queue supervision", causing packet failure on overflow and reducing throughput on underflow.

The "Weighted Fair Queuing (WFQ)" is a low-supported packet forecast algorithm that approximates "Generalized Processor Sharing" (GPS) [2], [19]. GPS supposes that infinitely divisible input traffic and all sessions will be serviced simultaneously. WFQ programmes packets through determining the virtual completion instance based on "arrival time", "size", and "related weights".

The scheduler determines the fundamental completion moment when the packet arrives at the queue. Where the essential completion moment indicates when the same packet will be processed. WFQ sorts packets in ascending order of essential completion instance. It ensures to facilitate every one flow has a bandwidth share proportionate to the assigned weight.

The "Weighted Process Queuing (WPQ)" rules provide QoS through providing reasonable (committed) bandwidth to the entire network traffic in support of controlling "packet loss", "latency", and "jitter". The "ToS field" in the "IP header" is used to recognize the weight. The "WFQ" rules determine the weight of traffic, so low-bandwidth traffic acquires high priority [10]. An exceptional characteristic of this queuing regulation is that instantaneous interactive traffic is progressed to the obverse of the queue and former bandwidth distributes move significantly between different flows as shown in Fig. 3.



Fig.3. Weighted Fairness Queue Control Mechanism

Weighted Fair Queuing (WFQ) [3] provides a process queue that divides the available bandwidth through traffic queues based on weights. Each flow, or set of it, is associated with an independent queue that is weighted so that critical traffic has a higher priority than less critical traffic. At a congested time, traffic in each queue (a single stream or a set of it) is fairly protected and handled

according to its weights.

Arrival packets are classified into different queues by examining the packet header fields, including attributes such as source and destination networks or MAC addresses, protocols, source and destination port and socket numbers of sessions, or "Diff-Serv-Code-Point" (DSCP) value. Each queue shares transport services relative to its associated weight. All traffic of the same class is treated imprecisely.

- *WEQ Scheduler*

In the communication network for data transmission, packets belonging to unlike traffic flows frequently share a link to the destination. If the node cannot transmit all the packets it receives at a particular instant, the packet queue is started. Therefore, the flow control procedure must deal with these queues by adjusting the instruct and quantity of data that each source be able to transmit.

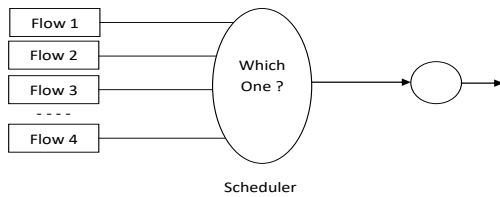


Fig.4. WFQ Scheduler

Fig. 4 shows a WFQ scheduler which selects the packets queue header and organizes it according to the packet weight assigned. Packet weight is assigned based on the data packet size. Let's assume a router has a capacity buffer of 1000 Mb data for transmitting. Scheduler control data inflow to less than 1000 Mb so that no data loss happens at the router and it should congest. To prepare the router data packet from the flow it picks 3 packets from each queue and arranges them in increasing order of their weight. The sum of total packet size must be below router buffer capacity.

#### D. TFRC with WFQ Mechanism

This mechanism began by sending packets from the source to the destination. In the early stages, data is transmitted in an incremental manner, allowing for fast bandwidth utilization. Transmission packets received by the WFQ server classify packets, calculate weights, and estimate flow rates. When all queue buffers are filled, the last packet received is discarded. The packet deletion response is sent to the source. Invokes the TFRC congestion control mechanism and calculates a new baud rate based on the server's current RTT. In Fig. 5 describe the flow diagram of TFRC with WFQ.

Based on above flow mechanism it needs to estimate two factors. (1) *Estimation of Queue Flow Rate* and, (2) *TFRC estimation of Transmission Rate* for appropriate transmission as discussed below.

- *Estimation of Queue Flow Rate*

Let's assume a packet  $P$  arrives at time  $P_t$  and its data length is  $P_l$  for a queue  $q$ . Then the estimation flow rate  $E_r$  of queue  $q$  will be,

$$E_{r\_new} = \left(1 - \frac{(P_t - P_{t-1})}{C}\right) \times \frac{P_l}{(P_t - P_{t-1})} + \frac{(P_t - P_{t-1})}{C \times E_{r\_old}} \quad (1)$$

where  $P_t$  is current packet arrived time,  $P_{t-1}$  is the previous packet arrived time,  $E_{r\_old}$  is the last estimated rate and  $C$  is a constant.

If we have  $n$  queue each queue have its own  $E_r$ , the then  $q_n$  which have high  $E_r$  will be selected for updating. The updating also depends on the queue is congested or not. We assume that a queue is considered congested if  $E_r \geq R(t)$ , where  $R(t)$  is the speed of the router which linked to WFQ server. In that sense, if the rate of arrival less than  $R(t)$  no packet is dropped, but if  $E_r \leq R(t)$  then,  $E_r - R(t) / E_r$  number of the packets will be dropped to maintain the flowing smoothness and avoid congestion. This provides an effective way of packet transmission with fair sharing of bandwidth. Every time this degradation occurs, the arrival rate of the incoming from the source decreases, and the rate of decrease is estimated by the TFRC at each source node.

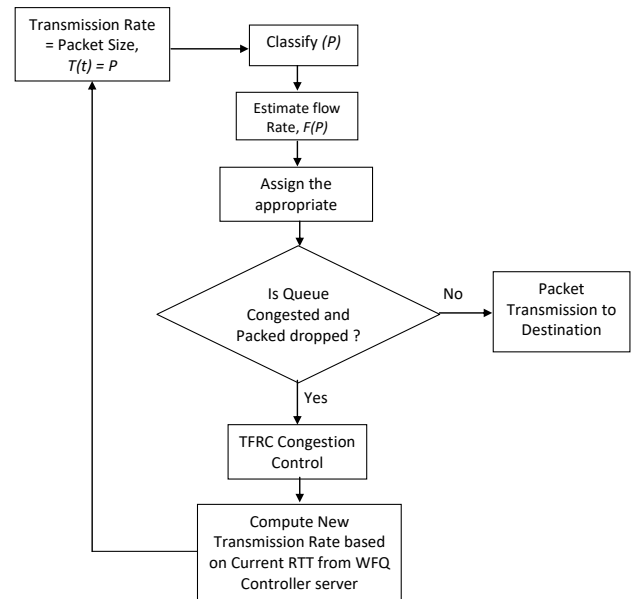


Fig.5. TFRC with WFQ Flow Diagram

- *TFRC Estimation of Transmission Rate*

One of the major problems facing the TFRC protocol to approximation the TFRC communication rate is related to the response capability requirement with approved communication rate. This is in reverse comparative to the computed "RTT". If the conferences contribute to the equivalent traffic jam, it expects the same bandwidth share. Thus achieving the identical communication rate. Adversely, this is denial in the condition, if RTT is very different between meetings. The entire "RTT" of a network capable of accessible as follows,

$$RTT_{Tot} = RTT_Q + RTT_C + RTT_T + RTT_M$$

where  $RTT_{Tot}$  is the entire RTT of a network,  $RTT_Q$  is the "RTT" due to a queuing delay,  $RTT_C$  is the RTT occurs due to computational delay,  $RTT_T$  is the RTT due to

transmission delay, and  $RTT_M$  is an RTT due to the movement delay. The advancement of device development makes  $RTT_C$  and  $RTT_T$  delay almost negligible. So, the current total RTT is computed based on the summation of  $RTT_Q$  and  $RTT_M$  as,

$$RTT_{Tot} = RTT_Q + RTT_M$$

The main cause of transmission rate reduction is due to the RTT-unfairness which is caused due to mainly  $RTT_Q$  and  $RTT_M$  and it's computed using the equation(2) give below.

$$T = \frac{s}{R\sqrt{2bp/3} + t_{RTO}(3\sqrt{3bp/8})p(1+32p^2)} \quad (2)$$

where,  $T$  is the "transmit rate in bytes/s",  $s$  is the "packet size in bytes",  $R$  is the "round-trip time",  $p$  is the "steady state loss event rate",  $t_{RTO}$  is the "TCP retransmission timeout", and  $b$  is the "number of packets acknowledged by a single TCP acknowledgment". For the assessment of  $b$ , variously delayed acknowledgments are not implemented by TCP and therefore  $b = 1$ . The unfairness of RTT highly impact the transmission rate  $T$ , the proposed integrated approach reduces the RTT delay through WFQ to enhance the performance of TRFC data transmission. The experimental evaluation of the proposed work is discussed in the following section.

#### IV. EXPERIMENT EVALUATION

To estimate the development of the recommended process, we configured the experimental setup on the "GloMoSim network simulator". We carry out experiments for TRFC traffic and also for the proposed TRFC with WFQ mechanism. The network configured for the simulation is shown in Fig. 6. The performance of TRFC with WFQ is evaluated to the "TRFC" to illustrate the improvisation in the high bandwidth-delay creation atmosphere. We define the network bandwidth to 1Gbps and RTT to 100 ms.

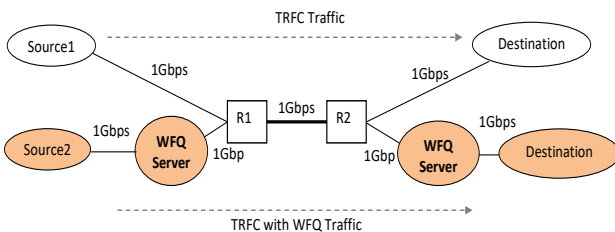


Fig.6. Simulation Network Configured

The simulation is evaluated varying different data loads from source to destination variation from 20-100pkts/s, and size of each packet is 1024bytes (1Mb). This objective of this load variation to create a congestion environment to evaluate the efficiency of the proposal. We measure Throughput, Packet Drops and Transmission Delay metric to evaluate the performance.

In Fig. 7 shows the throughput comparison between

TFRC and TFRC+WFQ Traffic. TFRC+WFQ shows an improvisation over TFRC in comparison to throughput due to the efficient congestion handling and keeping a smooth data flow rate. Whereas the TFRC exponentially increases the transmission rate in the beginning to achieve more than 90% throughput but once it reaches to maximum bandwidth limit or a packet loss occurs, it reduces its transmission rate to maintains the flowing smoothness which reduces its throughput with increased offer loads. In the case of TFRC+WFQ it manages the data inflow in fairness queue efficiently to achieve more than 98%, throughput, but with high traffic load it also its shows few packet loss due to queue buffer limit cause failing of throughput. The comparison shows an average improvisation of 30% in comparison to TRFC.

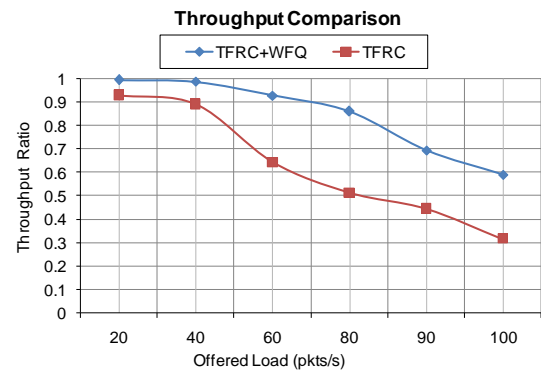


Fig.7. Throughput comparison between TFRC and TFRC with WFQ

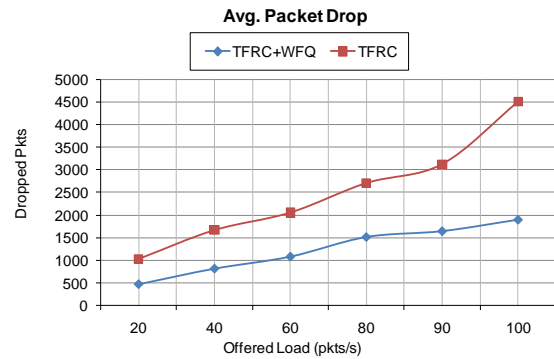


Fig.8. Avg. Packet Drop comparison between TFRC and TFRC+WFQ



Fig.9. Transmission Delay comparison between TFRC and TFRC+WFQ

In Fig. 8 and 9, shows the average packet drop and transmission delay comparison between TFRC and TFRC+WFQ. The increase in offered loads increases the



traffic cause congestion. TFRC initially increase its transmission rate to maintain traffic inflow, but the limitations of bandwidth slowly increase the RTT and the same time it may cause packet drop. Whereas, a TFRC + WFQ cause elevates the congestion through maintaining its WFQ and scheduling its transmission, which reduces the packet drops and also transmission delay. To minimize the packet loss TRFC reduces its transmission rate which affects its throughput. In the case of TFRC+WFQ packet loss is identified at WFQ server itself which provide low RTT in compare to TFRC destination feedback RTT. This makes an improvisation over TFRC in minimizing the packet drops and transmission delay.

## V. CONCLUSION

TFRC can perform TCP-friendly operations and balance it to meet some QoS constraints. However, TFRC has problems with high traffic speed and transmission delay by suppressing bandwidth and transmission rate. In this paper, we propose an integrated TFRC using WFQ approach to managing congestion in case of high traffic to improve TFRC performance. TFRC with WFQ mechanism is based on buffer queue management, which reduces router congestion through proper scheduling of data packets based on packet weights. Experimental evaluation of proposals shows instantaneous processing with low packet loss and transmission delay compared to TRFC. This mechanism reduces transmission rate suppression even in the case of TFRC's long RTT problem and packet loss.

## REFERENCES

- [1] M. C. Dias, L. A. Caldeira, and Perkusich, "A Traditional TCP Congestion Control Algorithms Evaluation in Wired-cum-Wireless Networks", IEEE Conference, DOI: 978-1-4799-5344-8/15, 2015.
- [2] K. Govardhan Reddy, P.V.S. Srinivas, and K. C. Sekharaiah, "Self-congestion prediction algorithm for efficient routing in Mobile Ad Hoc Network", International Conference on Control, Instrumentation, Communication and Computational Technologies 2014: 745-750.
- [3] N. Sivasubramaniam and P. Senniappa, "Enhanced Core-Stateless Fair Queuing with Multiple Queue Priority Scheduler", The International Arab Journal of Information Technology, Vol. 11, No. 2, March 2014.
- [4] S. Floyd, M. Handley, J. Padhye and J. Widmer, "TCP friendly rate control (TFRC): protocol specification", RFC-3448, Proposed Standard, September 2008.
- [5] I. Ullah, Zawar Shah and Adeel Baig, "S-TFRC: An Efficient Rate Control Scheme for Multimedia Handovers", Computer Science and Information Systems 13(1):45-69, 2015.
- [6] P. L. Utsav and S. S. Patil, "Extension of TCP To Improve The Throughput", International Conference on Computing Communication Control and Automation, 2015.
- [7] S.R.M. Krishna, M.N. Seeta Ramanath, V. Kamakshi Prasad, "Optimal Reliable Routing Path Selection in MANET through Novel Approach in GA", International Journal of Intelligent Systems and Applications (IJISA), Vol.9, No.2, pp.35 - 41, 2017. DOI: 10.5815/ijisa.2017.02.05
- [8] D. Murray and Kevin Ong, "Experimental Evaluation of Congestion Avoidance", IEEE 29th International Conf. on Adv. Information Networking and Applications, 2015.
- [9] M. Scharf, "Performance Analysis of the Quick-Start TCP Extension", In IEEE Fourth International Conference on Broadband Communications, Networks and Systems, pp. 942-951, 2007.
- [10] K. Govardhan Reddy, P. V. S. Srinivas, C. Sekharaiah K., "Congestion Avoidance and Data Prioritization Approach for High Throughput in Mobile Ad Hoc Networks" International Journal of Applied Engineering Research, Volume 10, Number 4 (2015), pp.8469-8486
- [11] S. V. Mallapur, S. R. Patil, J. V. Agarkhed, "A Stable Backbone - Based on Demand Multipath Routing Protocol for Wireless Mobile Ad Hoc Networks", International Journal of Computer Network and Information Security(IJCNIS), Vol.8, No.3, pp.41 - 51, 2016.DOI: 10.5815/ijcnis.2016.03.06
- [12] S. Ha, I. Rhee, and L. Xu, "Cubic: a new TCP-friendly high-speed TCP variant", SIGOPS Oper. Syst. Rev., vol. 42, pp. 64-74, July 2008.
- [13] A. Sathiseelan and G. Fairhurst, "TCP-Friendly Rate Control (TFRC) for bursty media flows", In Elsevier Journal on Computer Communications Vol-34, 1836-1847, 2011.
- [14] S. Lee, H. Roh, H. Lee, and K. Chung, "Enhanced TFRC for High-Quality Video Streaming over High Bandwidth-Delay Product Networks", Journal Of Communications And Networks, Vol. 16, No. 3, June 2014.
- [15] G. Sharma, M. Singh, P. Sharma, "Modifying AODV to Reduce Load in MANETs", International Journal of Modern Education and Computer Science(IJMECS), Vol.8, No.10, pp.25 - 32, 2016.DOI: 10.5815/ijmeecs.2016.10.04
- [16] Cisco Systems, "Bandwidth management", Cisco Visual Networking Index: Forecast and Methodology, May 2012.
- [17] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, 1(4): 397-413, 1993.
- [18] G. Hasegawa, M. Murata, and H. Miyahara, "Fairness and stability of the congestion control mechanisms of TCP", Telecommunication Systems, Vol. 15, 167-184, 2000.
- [19] M. Elshaikh, M. Othman, S. Shamala, and J. Desa, "A New Fair Weighted Fair Queuing Scheduling Algorithm in Differentiated Services Network", International Journal of Computer Science and Network Security, vol. 6, no. 11, pp. 267-271, 2006.
- [20] M. A. Talaat, M. A. Koutb, and H. S. Sorour, "ETFR: Enhanced TFRC for media traffic over the Internet", Int. Journal Comput. Netw., pp. 167-177, 2011.
- [21] D. Liu, M. Allman, S. Jin, and L. Wang, "Congestion control without a startup phase", In Proc. PFLDNet Workshop, 2007.
- [22] H. N. Elmahdy and Mohamed H. N. Taha, "The Impact of Packet Size and Packet Dropping Probability on Bit Loss of VoIP Networks", Proceeding in ICGST-CNIR Journal, Egypt, Vol-8, Issue-2, January 2009.
- [23] S. Floyd, Handley M, J. Padhye, J. Widmer, "Equation-based congestion control for unicast applications", In Proceedings of ACM SIGCOMM, Sweden, pg 43-56,

- 2000.
- [24] S. Maamar, B. Abderezzak, "Predict Link Failure in AODV Protocol to provide Quality of Service in MANET", International Journal of Computer Network and Information Security(IJCNIS), Vol.8, No.3, pp.1 - 9, 2016.DOI: 10.5815/ijcnis.2016.03.01
- [25] B. Dekeris, T. Adomkus and A. Budnikas, "Analysis of QoS Assurance Using Weighted Fair Queueing (WFQ) Scheduling Discipline with Low Latency Queue (LLQ)", Information Technology Interfaces ITI, Cavtat, 2006.
- [26] M. Georg, C. Jechlitschek, and S. Gorinsky, "Improving Individual Flow Performance with Multiple Queue Fair Queueing", in Proceedings of the 15th IEEE International Workshop on QoS, Evanston, USA, pp. 141-144, 2007.
- [27] D. Hang, H. Shao, W. Zhu, and Y.Q. Zhang, "TD2FQ: An integrated traffic scheduling and shaping scheme for DiffServ networks", In IEEE International Workshop on High-Performance Switching and Routing, 2001.
- [28] I. Hwang, B. Hwang, and C. Ding, "Adaptive Weighted Fair Queueing with Priority (AWFQP) Scheduler for Diffserv Networks", Journal of Informatics & Electronics, vol. 2, no. 2, pp. 1519, 2008.
- [29] N. Josevski and N. Chilamkurti, "A Stabilized Core-Stateless Fair Queueing Mechanism for a Layered Multicast Network", In Proceedings of International Conference on Networking and Services, California, USA, pp. 16-21, 2006.
- [30] M. F. Homg, W.T. Lee, K.R. Lee and Y.H. Kuo, "An Adaptive Approach to Weighted Fair Queue with QoS Enhanced on IP Network", IEEE Catalogue No. 01, CH37239, 2001.
- [31] D. Sisalem and H. Schulzrinne, "The loss-delay based adjustment algorithm: A TCP-friendly adaptation scheme", In Proc. NOSSDAV, pp. 215-226 1998.
- [32] S. Floyd, "Limited slow-start for TCP with large congestion windows", RFC 3742, 2004.

### Authors' Profiles



**Y.Narasimha Reddy**, Research Scholar, Rayalaseema University, Kurnool working as Asst. Professor in Brindavan Institute of Technology & Science, Kurnool in the department of CSE. Certified professional of CCNA and CCAI by CISCO, Bangalore. Areas of Interest are Wireless Ad-Hoc networks, Sensors Networks, IOT and Java. Present research work is on "Congestion Control in Wireless AdHoc networks" under the guidance of Dr.Podili.V.S.Srinivas, Professor, Sreenidhi Institute of Technology and Science, Hyderabad.



**Dr. P V S. Srinivas**, Professor, Sreenidhi Institute of Technology and Science, Hyderabad. He has an extensive experience in teaching and as an administrator. He is very intense and dedicated as a Teacher and Mentor. He had 25 years of experience. In the industry for 2 years and 23 years in academics. His main research areas include Computer Networks, Cloud Computing, and IoT. He published technical papers in peer-reviewed and globally reputed journals. He is a member of MIEEE, FIE, CSI, ISTE, IAENG.

**How to cite this paper:** Y. Narasimha Reddy, P V S. Srinivas, "A Combined TCP-friendly Rate control with WFQ Approach for Congestion Control for MANET", International Journal of Computer Network and Information Security(IJCNIS), Vol.10, No.6, pp.52-59, 2018.DOI: 10.5815/ijcnis.2018.06.05