

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

## TCP in MANET : Challenges and Solutions

Kaushika Patel <sup>1</sup>, Nayana Ram <sup>2</sup>, Virendra Barot <sup>3</sup>

Assistant Professor, Department of Electronics, BVM Engineering College, V.V.Nagar, Gujarat, India <sup>1</sup>

P.G. Student, Department of Information Technology, SVIT, Vasad, Gujarat, India <sup>2</sup>

Assistant Professor, Department of Information Technology SVIT, Vasad, Gujarat, India <sup>3</sup>

**ABSTRACT:** TCP (Transmission Control Protocol) is a reliable transmission control protocol. One of primary elements of TCP is to transmit information in a secured way over a system. . To this end, TCP utilizes a congestion control mechanism. In the TCP congestion control mechanism, congestion occurring during data transmission is detected from a data packet loss to directly control a data throughput at a sender side and lost data packets are then retransmitted. The important problem in most of wireless networks like mobile ad hoc networks is that TCP was originally designed for wired networks, but has subsequently been used for wireless networks. So when TCP is utilized with remote systems like MANET there are numerous issues identified with execution. A Mobile Ad hoc Network is (MANET) a constantly self-arranging, framework less system of cell phones associated without wires. In writing numerous arrangements are proposed to take care of the issues and enhance the productivity of TCP in MANET. So in the wake of concentrating on distinctive arrangement in this paper we have given review of diverse systems and plans proposed for TCP in MANET.

**KEYWORDS:** TCP, MANET, Tahoe, Reno, New-Reno, SACK, TCP Westwood, TCP Westwood-NR, Congestion control.

### I. INTRODUCTION

A Mobile Ad hoc Network is (MANET) a continuously self-configuring, infrastructure-less network of mobile devices connected without wires. Infrastructure is unavailable in the situations like Military war fields, disaster relief, sensor networks, Wireless mesh network etc. Every node in a MANET is allowed to move autonomously in any heading, and will in this way change its connections to different nodes much of the time. In these networks, congestion occurs in any intermediate node when data packets travel from source to destination and they incur high packet loss and long delay, which cause the performance degradations of a network [4]. Setting up of fixed access points and backbone infrastructure is not always viable because Infrastructure may not be present in a disaster area or war zone and for short range radios like ; Bluetooth (range ~ 10m) infrastructure may not be practical. For this situation, a MANET driven by battery force can be immediately conveyed to set up a system domain.

#### 1.1 Mobile Ad hoc Network (MANET)

In MANET when bundle drop arrives it is because of connection disappointment brought about by node portability however TCP considers it as blockage and applies it congestion control mechanism. In view of this issue the bandwidth can't be use completely. Such a variety of scrutinizes have been led to enhance the execution of TCP.

Routing is another territory of exploration in MANET. Ad-Hoc network routing protocols are commonly divided into three main classes; Proactive, reactive and hybrid protocols.

1) Proactive Protocols: In proactive routing, each node has to keep up one or more tables to store routing data, and any adjustments in system topology should be reflected by propagating all updates throughout the system in order to keep up a reliable system view. Sample of such schemes are the conventional routing schemes: Destination sequenced distance vector (DSDV). They attempt to maintain consistent, progressive routing data of the entire system. It minimizes the delay in communication and permit nodes to rapidly figure out which nodes are available or reachable in the system.

## International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

2) Reactive protocols: Reactive routing is otherwise called on-demand routing protocol since they don't keep up routing data or routing action at the system nodes preceding to communication. On the off chance that a node needs to send a packet to another node then this protocol scans for the route in an on-demand way and builds up the connection so as to transmit and get the packet. The route discovery happens by flooding the route ask for packets all through the system. Illustrations of reactive routing protocols are the Ad-hoc On-demand Distance Vector routing (AODV) and Dynamic Source Routing (DSR).

3) Hybrid Protocols: They presents a hybrid model that joins receptive and proactive routing protocols. The Zone Routing Protocol (ZRP) is a hybrid routing protocol that partitions the system into zones. ZRP gives a various levelled architecture where every nodes needs to keep up extra topological data requiring additional memory.

Numerous examinations have been done to enhance the execution of TCP at network layer by enhancing the routing strategy. In this way, there is extension to enhance the execution of TCP at transport layer. So next , presentation of TCP in MANET is given.

### II. TRANSMISSION CONTROL PROTOCOL (TCP)

TCP is a protocol used for transmission of data between nodes connected to a network together with IP (Internet Protocol). One of primary elements of TCP is to transmit information in a secured way over a system. To this end, TCP utilizes a congestion control mechanism. In the TCP congestion control mechanism, congestion occurring during data transmission is detected from a data packet loss to directly control a data throughput at a sender side and lost data packets are then retransmitted. FIG. 1 illustrates a change in the throughput depending on a conventional Slow-Start/Congestion Avoidance algorithm [21].

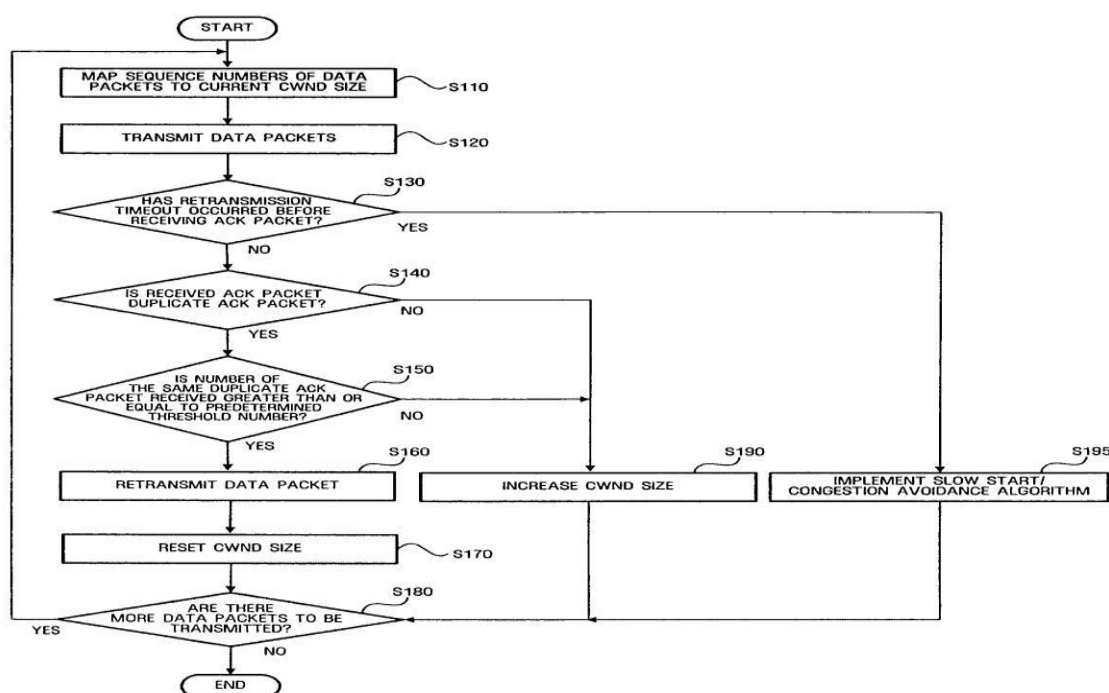


FIG. 1 : Conventional Slow-Start/Congestion Avoidance algorithm[21]

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

A congestion window (abbreviated as "CWND") size is initialized to one TCP segment at the time when the TCP of a sending network apparatus initiates transmission of data packets. The TCP segment is incremented by one (1) for every acknowledgement (ACK) packet received from a receiving network apparatus. Accordingly, the CWND size is increased exponentially, as labelled **110** in FIG. 1, which is called a slow-start algorithm[21].

However, if an ACK packet for the transmitted data packet is not received, the TCP waits for a retransmission timeout, labeled **170**. If no ACK packet is received until the retransmission timeout **170** occurs, the TCP considers that congestion occurs in the network. In this case, the TCP sets a slow-start threshold (SSTHRESH) to a value SSTHRESH1, labeled **150**, the value being a half the CWND size set immediately before the retransmission timeout. In addition, the TCP sets the CWND size to one (1) TCP segment and transmits data packets subsequently transmitted after the lost data packet using a slow-start algorithm, labeled **120**[21].

If the CWND size reaches the SSTHRESH1 **150** during transmission using the slow-start algorithm, the TCP performs transmission using a congestion avoidance algorithm, labelled **130**, in which CWND is linearly increased, in consideration of congestion of the network. Thereafter, if no ACK packet is received once again and there occurs a retransmission timeout, labelled **180**, the TCP sets the slow-start threshold (SSTHRESH) to a value SSTHRESH2, labelled **160**, the value being a half the CWND size set immediately before occurrence of the retransmission timeout, sets the CWND size to 1 TCP segment, and a low-start algorithm, labelled **140**, is started again[21].

Congestion Control is the heart of TCP. To enhance the execution of TCP in remote systems like MANET different congestion control policies for TCP variations are given which are talked about next.

## 2.1 Tahoe TCP

Current TCP executions contain various algorithms went for controlling system congestion while keeping up good user throughput. Early TCP executions took after ago-back- N model utilizing cumulative positive acknowledgement and requiring a retransmit timer expiration to re-send information lost during transport. These TCPs did little to minimize system congestion. The Tahoe TCP execution included various new algorithms and refinements to prior implementation. The new algorithms incorporate Slow-Start, Congestion Avoidance, and Fast Retransmit [14]. The refinements incorporate a change to the round-trip time estimator used to set retransmission timeout values. The sum total of what alterations have been depicted somewhere else [14,15]. The Fast Retransmit algorithm is of exceptional enthusiasm for this paper in light of the fact that it is changed in consequent variants of TCP. With Fast Retransmit, in the wake of accepting a little number of duplicate acknowledgement for the same TCP fragment (dup ACKs), the information sender deduces that a packet has been lost and retransmits the packer without waiting for a retransmission timer to expire, prompting higher channel usage and connection throughput[10].

## 2.2 Reno TCP

The Reno TCP execution held the improvements joined into Tahoe, yet altered the Fast Retransmit operation to incorporate Fast Recovery [11]. The new algorithms keeps the communication way ("pipe") from going void after Fast Retransmit, in this way avoiding need to Slow-Start to re-fill it after a single packet loss. Fast Recovery works by assuming each dup ACK got speaks to a single packet having left the pipe. In this manner, amid Fast Recovery the TCP sender has the capacity make keen assessments of the amount of outstanding data. Fast Recovery is entered by a TCP sender subsequent to accepting a initial threshold of dup ACKs. This threshold, normally known as *tcprexmthresh*, is by and large set to three. Once the threshold of dup ACKs is gotten, the sender retransmits one packet and decreases its congestion window by one half. Rather than slow- starting, as is performed by a Tahoe TCP sender, the Reno sender utilizes extra incoming dup ACKs to clock subsequent outgoing packets.

In Reno, the sender' s usable window gets to be  $\min (awin, cwnd+ndup)$  where *awin* is the receiver's advertised window, *cwnd* is the sender' s congestion window, and *ndup* is kept up at 0 until the number of dup ACKs comes to *tcprexmthresh*, and from that point tracks the number of duplicate ACKs. Hence, amid Fast Recovery the sender "inflates" its window by the number of dup ACKs it has gotten, by perception that each dup ACK shows some packets has been expelled from the system and is currently cached at the receiver. After entering Fast Recovery and retransmitting a single packet, the sender viably holds up until a half a window of dup ACKs have been gotten, and after that sends another packet for each extra dup ACK that is gotten. Upon receipt of an ACK for new information (called a "recuperation ACK"), the sender leaves Fast Recovery by *ndup* setting to 0. Reno's Fast Recovery algorithm

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

is upgraded for the situation when a single packet is dropped from a window of information. The Reno sender retransmits at most one dropped packet for each round-trip time. Reno essentially enhances the conduct of Tahoe TCP when a single packet is dropped from a window of information, yet can experience the ill performance issues when various packets are dropped from a window of data[10].

## 2.3 New-Reno TCP

The New-Reno TCP incorporates a little change to the Reno algorithm at the sender that dispenses with Reno's sit tight for a retransmit timer when multiple packets are lost from a window [12,13]. The change concerns the sender's conduct amid Fast Recovery when a partial ACK is gotten that acknowledges some yet not all of the packets that were remarkable toward the begin of that Fast Recovery period. In Reno, partial ACKs take TCP out of Fast Recovery by "deflating" the usable window back to the size of the congestion window. In New-Reno, partial ACKs don't take TCP out of Fast Recovery. Rather, partial ACKs got amid Fast Recovery are dealt with as a sign that the packets quickly taking after the acknowledged packets in the sequence space has been lost, and ought to be retransmitted. In this manner, when various packets are lost from a single window of information, New-Reno can recuperate without a retransmission timeout, retransmitting one lost packet for each round-trip time until all the the lost packets from that window have been retransmitted. New-Reno stays in Fast Recovery until all the information exceptional when Fast Recovery was started has been acknowledged [10].

## 2.4 SACK TCP

The SACK alternative takes after the arrangement in [16]. From [16], the SACK alternative field contains various SACK blocks, where every SACK block reports a non-contiguous set of information that has been gotten and queued. The first block in a SACK alternative is required to report the information recipient's most as of late got segment, and the extra SACK blocks rehash the most as of late reported SACK blocks [16]. In these simulations every SACK alternative is accepted to have space for three SACK blocks. At the point when the SACK alternative is utilized with the Timestamp choice determined for TCP Extensions for High Performance [17], then the SACK choice has space for just three SACK blocks [16]. In the event that the SACK choice were to be utilized with both the Timestamp choice and with T/(TCP Extensions for Transactions) [18], the TCP alternative space would have space for just two SACK blocks.

The congestion control algorithms actualized in SACK TCP are a preservationist expansion of Reno's congestion control, in that they utilize the same algorithms for expanding and diminishing the congestion window, and roll out insignificant improvements to the next congestion control algorithms. Adding SACK to TCP does not change the fundamental underlying congestion control algorithms. The SACK TCP usage protects the properties of Tahoe and Reno TCP of being hearty in the presence of out-of-request packets, and uses retransmit timeouts as the recovery technique for final resort. The fundamental contrast between the SACK TCP execution and the Reno TCP implementation is in the conduct when multiple packets are dropped from one window of information.

As in Reno, the SACK TCP usage enters Fast Recovery when the information sender gets *tcprexmtthresh* duplicate acknowledgement. The sender retransmits a packet and slices the congestion window down the middle. Amid Fast Recovery, SACK keeps up a variable called pipe that speaks to the assessed number of packets outstanding in the path. (This contrasts from the mechanisms in the Reno execution.) The sender just sends new or retransmitted information when the assessed number of packets in the path is less than the congestion window. The variable pipe is augmented by one when the sender either sends new packet or retransmits an old packet. It is decremented by one when the sender gets a dup ACK packet with a SACK choice reporting that new information has been gotten at the receiver[10].

## 2.5 TCP Westwood

In TCP Westwood the sender constantly computes the connection Bandwidth Estimate (BWE) which is characterized as the offer of bottleneck bandwidth utilized by the connection. Along these lines, BWE is equivalent to

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

the rate at which information is conveyed to the TCP receiver. The estimate is depends on the rate at which ACKs are gotten and on their payload. After a packet misfortune sign , (ie, reception of 3 duplicate ACKs, or timeout expiration). , the sender resets the congestion window and the slow start threshold taking into account BWE. All the more definitely,  $cwin=BWE \times RTT$ [19].

## 2.6 TCP Westwood-NR

The issue with TCP Westwood is that it excludes the router's buffer size. Accordingly, alterations done to execute TCP Westwood-NR are practically identical to the ones actualized in the move from TCP Reno to TCP New reno. TCP WNR was expected to enhance execution under irregular or sporadic losses. This rendition was tried through simulation and demonstrated impressive addition as far as through-put in all situations.

## III.ISSUES OF MANET WITH TCP

TCP is a reliable connection oriented end-to-end convention which is utilized as a part of MANET for congestion control. Principle issue in TCP is that the sender expects that packet loss is happened in view of system congestion. Yet, this is not generally valid for MANETs. Since in MANET the packet drop is for the most part on account of connection failure not in view of congestion. Be that as it may, TCP takes it as congestion and applies its congestion control mechanism and declines the congestion window. So the most extreme data transmission cannot be use. In MANET, packets may be lost because of multipath fading, Doppler shift, buffer overflow, link-layer contention and channel errors. Be that as it may, the current TCP algorithms consider packet loss as congestion because of buffer overflow and call congestion control strategy independent of the above cases.

The Transmission Control Protocol (TCP) was initially intended for wired systems, yet has therefore been utilized for remote systems. For remote systems, throughput is a component of the window size parameter, which generally characterizes the number of bytes that can be sent at one time, and the Round Trip Time (RTT) for a packet. For instance, most extreme throughput may be equivalent to  $(MAX \text{ Window Size})/RTT$ . As remote systems keep on offering bigger peak data rates, a requirement for a bigger TCP window size exists. Then again, a bigger window size may bring about wasted radio resources.

In traditional telecommunication or data communication systems, conveying nodes build up end-to-end sessions using conventions such as TCP. At the point when a source node transmits a packet to a destination node, it likewise sets a clock to open an acknowledgment period, in light of a TCP-particular algorithms for packet traffic attributes, amid which the destination node must acknowledge right receipt of the packet. On the off chance that the source node does not get an acknowledgment from the destination node before the clock lapses, the timer issues a time out, the packet is resolved to be lost, and the source node must retransmit the packet to the destination node[22].

A lot of research is being carried out in the area of congestion control, routing of packets, modification of standard TCP protocol, designing of new routing protocol exclusively for MANET.

## IV.RELATED WORK

Jiwei Chen and Mario Gerla say TCP does not always get throughput gain by delaying unlimited acks. The maximal TCP throughput is achieved at a certain delay window that balances decreasing ack flow and burst loss. Thus, we introduced two novel adaptive delayed ack mechanisms compatible with TCP called TCP-DCA, based on the path hop length and/or end-to-end delay, to maximize the TCP performance for MANET and hybrid networks. Our proposed schemes are shown to have superior performance in static networks (wireless or hybrid wired/wireless networks). Jiwei Chen and Mario Gerla also demonstrated better TCP performance with our proposed mechanisms over different routing protocols in mobile ad hoc networks.

According to Zvi Rosberg and John Matthews although the congestion control of each TCP version aims to

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

achieve fair rates in isolation, the actual rates in a mixed environments (of UDP and multiple variants of TCP congestion controls) is far from being fair. Zvi Rosberg and John Matthews comprise a network rate management protocol (RMP) that controls the rate of all flows (at an aggregate level based on routes) subject to QoS requirements. The RMP control also facilitates a new TCP sliding-window congestion control based on the fair target rates computed by the RMP. Each non-TCP aggregate flow is policed by its respective edge router and each TCP flow adapts its window size as to achieve the RMP suggested fair target rate.

Author [3] says Pump slowly, fetch quickly (PSFQ) is a hop-by-hop downstream transport protocol which is reliable, scalable and robust. It uses NACK-based loss detection and notification, and local retransmission for loss recovery and is basically designed for retasking / reprogramming application which requires 100% packet reliability in downstream direction. Author provides an improved PSQF which works better than original PSFQ in terms of error tolerance and average latency.

T. Senthil Kumaran , V. Sankaranarayanan says in MANETs congestion occurs in any intermediate node when data packets travel from source to destination and they incur high packet loss and long delay, which cause the performance degradations of a network. [4] proposes an early congestion detection and adaptive routing in MANET called as EDAPR. Initially EDAPR constructs a NHN (non-congested neighbours) neighbours list and finds a route to a destination through an NHN node. All the primary path nodes periodically calculate its queue status at node level. While using early congestion detection technique, node detects congestion that is likely to happen and sends warning message to NHN nodes. The ancestor NHN node is aware of this situation and finds an alternate path to a destination immediately by applying adaptive path mechanism. Thus, EDAPR improves performance in terms of reducing delay, routing overhead and increases packet delivery ratio without incurring any significant additional cost.

According to Sangtae Ha and Injong Rhee Standard slow start does not work well under large bandwidth-delay product (BDP) networks. Sangtae Ha and Injong Rhee suggest a new slow start algorithm, called Hybrid Start (HyStart), that finds a “safe” exit point for slow start at which it can terminate and safely advance to the congestion avoidance phase without causing heavy packet loss. HyStart uses ACK trains and RTT delay samples to detect whether (1) the forward path is congested or (2) the current size of the congestion window has reached the available capacity of the forward path. HyStart is a plugin to TCP senders and requires no change on TCP receivers.

The primary objective of congestion control is to best utilize the available network resources and keep the load below the capacity says T. Senthilkumaran and V. Sankaranarayanan. The congestion control techniques to deal with TCP have been found inadequate to handle congestion in ad hoc networks, because ad hoc networks involve special challenges like high mobility of nodes and frequent changes of topology. Author proposes a method for dynamic congestion detection and control routing (DCDR) in ad hoc networks based on the estimations of the average queue length at the node level. Using the average queue length, a node detects the present congestion level and sends a warning message to its neighbors. The neighbors then attempt to locate a congestion-free alternative path to the destination. This dynamic congestion estimate mechanism supporting congestion control in ad hoc networks ensures reliable communication within the MANET. New delay-based congestion control mechanisms like Caia-HamiltonDelay(CHD) or Caia Delay-Gradient(CDG) are becoming popular due to their effectiveness even over wireless links according to Satoshi Utsumi and Salahuddin. CHD and CDG are designed to be have friendly with conventional TCP New Reno flows. However, due to their delay- based window update mechanisms ,CHD or CDG performance is often vulnerable to th eco-existing aggressive TCP New Reno flows. In addition ,in case of CHD ,it is practically very difficult to choose the optimum values of tuning parameters with a view to ensuring the expected performance.

Satoshi Utsumi and Salahuddin therefore propose a new mechanism to overcome the issues that is Wireless Friendly Congestion Control (WFCC). Which objective is to devise a congestion control scheme that is (i) friendly with TCP New Reno flows over wireless links when deployed together, (ii) free from delicate operational parameters and (iii) robust against link errors under a wide range of network buffer space says Satoshi Utsumi and Salahuddin.

Min Wang and Junfeng Wang observed that TCP variants could not be automatically chosen according to the lower network environments . So Min Wang and Junfeng Wang implement an adaptive congestion control framework (ACCF), which can automatically transition among existing congestion control mechanisms according to the change of the network status. It can switch the congestion control approaches between the delay-based ones and the loss-based ones according to the network status.

Shahin Mahdizadeh Aghdam and Mohammad Khansari propose a new content-aware cross layer WMSN Congestion Control Protocol (WCCP) by considering the characteristics of multimedia content. WCCP employs a

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

Source Congestion Avoidance Protocol (SCAP) in the source nodes, and a Receiver Congestion Control Protocol (RCCP) in the intermediate nodes. SCAP uses Group of Picture (GOP) size prediction to detect congestion in the network, and avoids congestion by adjusting the sending rate of source nodes and distribution of the departing packets from the source nodes. In addition, RCCP monitors the queue length of the intermediate nodes to detect congestion in both monitoring and event-driven traffics. Moreover, to improve the received video quality in base stations, WCCP keeps the I-frames and ignores the other less important frame types of compressed video, in the congestion situations says Shahin Mahdizadeh Aghdam and Mohammad Khansari.

## V. CONCLUSION

From the study about TCP in MANET it is reason that TCP is initially proposed for wired system so when it is utilized for remote system like MANET issues happens identified with its execution. Numerous variations have likewise been proposed to enhance the execution like Tahoe, Reno, New-Reno, SACK and TCP Westwood. In this paper diverse variations of TCP have been concentrated on. Likewise different methods and plans to enhance the execution of the TCP in MANETs are reviewed.

## VI. SCOPE OF IMPLEMENTATION

Congestion control algorithm is the primary piece of TCP. As remote systems keep on offering bigger peak data rates, a requirement for a bigger TCP window size exists. On the other hand, a bigger window size may bring about wasted radio resources. So we can attempt to enhance the CWND parameter of congestion control mechanism. The issue with TCP Westwood is that it excludes the router's buffer size. So there is degree to discover answer for enhance its execution. At the point when a source node transmits a packet to a destination node, it additionally sets a timer to open an acknowledgment period, taking into account a TCP algorithm for packet traffic characteristics, amid which the destination node must acknowledge right receipt of the packet. On the off chance that the source node does not get an acknowledgment from the destination node before the clock lapses, the clock issues a time out, the packet is resolved to be lost, and the source node must retransmit the packet to the destination node. In this manner, the RTT(Round Trip Time) can likewise be an essential parameter to enhance the execution of TCP. Work can likewise be done to attempt discover answer weather the packet loss is due to congestion or link failure which is the case of wireless networks.

## REFERENCES

- [1] Jiwei Chen , Mario Gerla, Yeng Zhong Lee, M.Y. Sanadidi "TCP with delayed ack for wireless networks" : Ad Hoc Networks 6 (2008) 1098–1116, Science Direct, 2007
- [2] Zvi Rosberg , John Matthews , Moshe Zukerman , "A network rate management protocol with TCP congestion control and fairness for all", Computer Networks 54 (2010) 1358–1374, 2009, Science Direct, 2009
- [3] Chirag K. Rupani , Trilok C. Aseri , "An improved transport layer protocol for wireless sensor networks" : Computer Communication 34(2011) 758-764, Science Direct , 2011
- [4] T. Senthil Kumaran , V. Sankaranarayanan, " Early congestion detection and adaptive routing in MANET": Egyptian Informatics Journal (2011) 12, 165–175, Science Direct, 2011
- [5] Sangtae Ha , Injong Rhee , "Taming the elephants: New TCP slow start": Computer Networks 55 (2011) 2092–2110 , Science Direct ,2011
- [6] T. Senthilkumaran a, V. Sankaranarayanan "Dynamic congestion detection and control routing in ad hoc networks" : Journal of King Saud University – Computer and Information Sciences (2013) 25, 25–34, Science Direct ,2013
- [7] Satoshi Utsumi , Salahuddin Muhammad Salim Zabir , "A new high-performance TCP friendly congestion control over wireless networks" : Journal of Network and Computer Applications 41 (2014) 369–378, Science Direct, 2014
- [8] Min Wang , Junfeng Wang, Sunyoung Han, " Adaptive congestion control framework and a simple implementation on high bandwidth-delay product networks" : Computer Networks 64 (2014) 308–321 , Science Direct , 2014
- [9] Shahin Mahdizadeh Aghdam , Mohammad Khansari , Hamid R. Rabiee , Mostafa Salehi , "WCCP: A congestion control protocol for wireless multimedia communication in sensor networks" :Ad Hoc Networks 13 (2014) 516–534 ,Science Direct ,2014
- [10] Kevin Fall and Sally Floyd , "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP"
- [11] V. Jacobson. "Congestion Avoidance and Control,". *SIGCOMM Symposium on Communications Architectures and Protocols*, pages 314–329, 1988. An updated version is available via <ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z>.
- [12] J. Hoe. "Start-up Dynamics of TCP's Congestion Control and Avoidance Schemes,". Jun. 1995. Master's thesis, MIT.
- [13] D.D. Clark and J. Hoe. "Start-up Dynamics of TCP's Congestion Control and Avoidance Schemes,". Technical report, Jun. 1995. Presentation to the Internet End-to-End Research Group, cited for acknowledgement purposes only.
- [14] V. Jacobson. "Congestion Avoidance and Control,". *SIGCOMM Symposium on Communications Architectures and Protocols*, pages 314–329, 1988. An updated version is available via <ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z>.



ISSN(Online): 2319-8753  
ISSN (Print): 2347-6710

## International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

**Vol. 4, Issue 12, December 2015**

- [15] W. Richard Stevens. "TCP/IP Illustrated, Volume 1: The Protocols"; Addison Wesley, 1994
- [16] Matthew Mathis, Jamshid Mahdavi, Sally Floyd, and Allyn Romanow. "TCP Selective Acknowledgment Options,". (Internet draft, work in progress), 1996.
- [17] D. Borman, R. Braden, and V. Jacobson. "TCP Extensions for High Performance,". Request for Comments (Proposed Standard) RFC 1323, Internet Engineering Task Force, May 1992. (Obsoletes RFC1185).
- [18] R. Braden. "T/TCP – TCP Extensions for Transactions Functional Specification,". Request for Comments (Experimental) RFC 1644, Internet Engineering Task Force, July 1994.
- [19] Mario Gerla, M. Y. Sanadidi, Ren Wang, and Andrea Zanella, Claudio Casetti, Saverio Mascolo, "TCP Westwood: Congestion Window Control Using Bandwidth Estimation"; Global Telecommunications Conference, GLOBECOM '01. IEEE , 1698 - 1702 vol.3,2001
- [20] Lalit Ratilal Kotecha, Raafat Edward Kamel, Sergio Aguirre," TCP window size performance optimization in wireless networks"; US 8639835 B2, 28 Jan 2014
- [21] Jin-hyun Sin, "Method for transmitting data in mobile ad hoc network and network apparatus using the same"; US7835273 B2, 16 Nov 2010
- [22] Samuel M. Levenson, Yutal T. Koh, Mark S. Hansen," Time out threshold shaping for wireless TCP communications";US 6791945 B1,14 Sep 2004