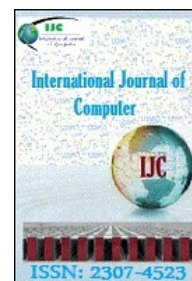




International Journal of Computer (IJC)

ISSN 2307-4531

<http://gssrr.org/index.php?journal=InternationalJournalOfComputer&page=index>

Implementation of TCP Congestion Control mechanism for Wireless Networks using TCP Reserved Field and Signal to Noise Ratio (SNR)

D. Hema Latha^{a*}, Prof. T. Venkat Narayana Rao^b, D. Rama Krishna Reddy^c

^aAsst. Professor, Dept of Comp. Science, Osmania University College For Women, Koti, Hyderabad-39, AP, India

^bProfessor, C.S.E, Sreenidhi Institute of Science & Technology, Hyderabad, AP, India

^cAsst. professor, Dept of Mathematics, UCS, Osmania University, Tarnaka, Hyderabad-07, AP, India

^ahlatha059@gmail.com

^btvnr Bobby@yahoo.com

^cdrkreddymca@gmail.com

Abstract

TCP is the most popular and widely used network transmission protocol. All most 90% of the Internet connections make use of TCP for communication. TCP is reliable for wired networks and it considers all packet timeouts in wired networks as due to network congestion and not because of bit errors. However, TCP suffers from performance degradation over error-prone wireless links, as it has no technique to distinguish error deficits from congestion deficits, with networking becoming more divergent, with wired and wireless topologies. It considers all packet deficits are due to congestion and subsequently reduces the packet burst transmission, at the same time decreasing the network throughput. In this paper a new TCP congestion control mechanism is proposed that is suitable and applicable for wireless and also for wired networks and is capable of distinguishing congestion deficits from error deficits. The proposed technique uses the reserved field of the TCP header to indicate whether the connection established is over a wired or a wireless link.

* Corresponding author.

E-mail address: hlatha059@gmail.com.

Further, the proposed technique influences the usage of Signal to Noise Ratio (SNR) to discover the reliability of the link and determine whether to decrease the packet burst or retransmit the timed-out packet. Investigations performed, revealed that the proposed mechanism confirmed to function amend in circumstances where timeouts were due to error and not due to congestion. Further, the future work can be enhanced upon the proposed mechanism so that it can leverage Cyclic Redundancy Check(CRC) and Header Error Check(HEC) errors so that it can be properly determined the reason for initializing transmission timeouts in wireless networks.

Keywords – TCP Congestion, Wireless Network, Reserved Field, SNR

1. Introduction

Congestion develops [1] when the number of packets sent to any network is more than it can handle. Numerous algorithms have been developed since the dawn of computer networking to deal with network congestion. In fact, the simplest way to solve congestion is to employ the conversation packet in which the sender will not send the new packet until the previous one is successfully delivered to the receiver. For this reason, the TCP protocol, one of the core protocols of the Internet protocol suite, employs the concept of congestion control which dynamically controls the flow of packet inside a network, and prevents network performance break down [2].

In the early days of data communication, the problem was with congestion detection as acknowledgement option was not available and sender not receiving an acknowledgment from the receiver (referred to as timeout) does not mean that the packet was lost due to congestion but also to noise or error in the transmission wire. However, with the advancement of technologies, packet loss due to transmission error became relatively infrequent as most communication trunks and network infrastructures achieved a high-level of reliability and resilience. As a result, transmission timeouts in modern computer wired networks are 99% due to congestion [3].

The reliability of data transmission has led the TCP protocol to be optimized for wired networks in a sense that any packet loss is considered to be the result of network congestion and not the result of network errors. Algorithmically, when a timeout is detected, the TCP congestion control algorithm suddenly reduces the packet burst to decrease the network load and relieve from congestion. In effect, the TCP congestion algorithm cannot distinguish the deficits caused by congestion from the one caused by error. Once the deficit arises, TCP manages with a congestion event and reduces its congestion window size by half. This unnecessary slowdown reaction decreases the throughput of TCP and reduces the overall speed of the network.

For implementation the above specified behavior of the TCP congestion algorithm is acceptable in wired networks as packet timeouts are mostly caused by congestion. However, this is totally inappropriate in wireless networks as wireless links may experience more error bits and packet deficit because of interference, fading, signal hand-off, and other radio effects; consequently, packet deficit in wireless networks cannot be considered as a deficit due to congestion. As a result, the TCP instead of retransmitting the lost packets, it decreases the transmission of packet bursts. This problem is commonly known as the TCP performance problem over wireless

network, study and research has been done by many researchers for many years now [4]. The chief solution in solving this problem is to allow TCP to differentiate between timeouts caused due to congestion and those caused by errors and noise in the wireless channel.

In this paper the authors proposed a new TCP congestion control mechanism suitable and applicable for wireless and also for wired networks and are destined to distinguish congestion deficits from error deficits. The proposed mechanism uses the reserved bits of the TCP header to indicate whether the established connection is over a wired or a wireless link. Moreover, the scheme employs SNR (Signal-to-Noise) ratio to discover the reliability of the link and determine whether to decrease packet burst or to retransmit the timed-out packet. In concise, when a timeout occurs, the TCP protocol investigate the type of the connection, if it is wired, then packet deficit is due to congestion so the classical slow-start congestion control algorithm is implemented to decrease the congestion window size. Alternatively, if the connection is wireless and SNR ratio is less than 5dB, then packet deficit is due to error and thus the timed-out packet is retransmitted leaving the congestion window intact. If the connection is wireless and SNR ratio is greater than 5dB, then packet deficit is due to congestion so the congestion window size is reduced to slow down the burst of packets transmission.

2. TCP Congestion Control Algorithm

Transmission Control Protocol (TCP) is the most superior protocol used in computer networking and for Internet applications. TCP congestion control algorithm: In TCP, when a connection is established, an appropriate window size must be selected between source and destination devices. Actually, two windows are available: the receiver's window and the sender's congestion window. The number of bytes that can be transmitted by the sender is the minimum of these two windows. Hence, in case the receiver's window size is 16 KB, and the sender's congestion window is 8 KB, then the transmission would occur at 8 KB. In contrast, if the receiver's window size is 8 KB and the sender's congestion window is 16 KB, then the transmission would occur at 8 KB. The sender or source device computes its congestion window size by examining the attributes of the medium network such as delays, traffic, and bandwidth; whereas, the receiver or destination device computes its window size depending on its buffer size.

When a TCP connection is established, the sender initiates the congestion window to the size of the maximum segment available on the connection. It then sends one single maximum segment 'n'. If this segment is acknowledged by the receiver before the time out condition, the sender adds another segment magnifying the size of its congestion window twice and sends the two segments '2n'. As each of these segments is acknowledged by the receiver, the congestion window is magnified by one maximum segment increasing its previous size to twice. Essentially, each acknowledged packet burst increases the congestion window until either a timeout condition occurs or the maximum size of the receiver's congestion window is reached. In case of a timeout condition, the segment size is decreased by half and so for the congestion window. The concept behind this behavior is to maintain, for instance, a window size of 4 KB as long as it is acknowledged by the receiver. Once, timeout condition occurs, it is reduced to 2 KB to avoid congestion [5, 6]. Example for TCP congestion algorithm is shown in Fig. 1.

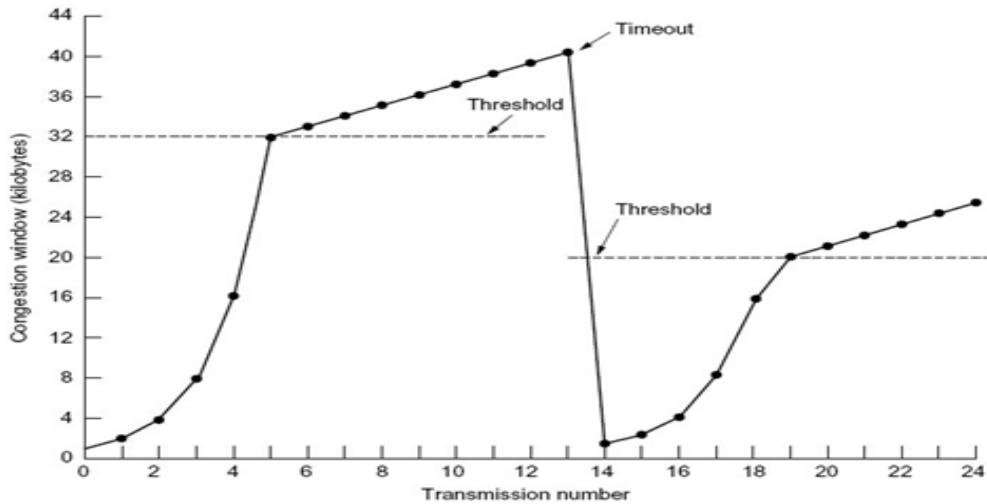


Figure 1. Example for the TCP Congestion Algorithm

3. Problem Statement - TCP over Wireless Network

Presently, all accomplishments of the TCP congestion algorithm assume that the timeouts are caused by congestion, and not by transmission errors. This conception works when implemented on wired networks because they are relatively reliable and exhibit very low or no bit errors. But, this conception does not work for wireless connections as they suffer from high error and packet loss rates; and hence, considered unreliable. For this reason, any packet loss in wireless transmission is by mistake considered by the TCP protocol as due to congestion which triggers the congestion algorithm to reduce the window size to one segment and consequently reducing transmission speed and packet throughput. For instance, if 25% of all packets are lost, then when the source transfers 200 packets per second, the throughput is 150 packets per second. If the sender slows down to 100 packets per second, the throughput is deteriorated up to 75 packets per second [7]. Initially the TCP congestion algorithm was not designed to support the error-prone wireless network, was only for very reliable wired network, it was impossible for the sender to differentiate between congestion deficit and error deficit. As a result, in timeout situations over wireless networks, the TCP often makes the wrong decision by slowing down the packet burst instead of retransmitting lost packets.

4. Correlated Work

Various strategies and approaches were extensively studied and experimented to solve the TCP congestion problem over wireless networks. Some of the most successful schemes are: I-TCP, Snoop, ECN, WTCP, Westwood, TCP Vegas, TCP Veno, M-TCP, and JTCP.

4.1. I-TCP

The Indirect-TCP (I-TCP) [8] is refinement of the original TCP protocol in which the network link between source (sender) and receiver (destination) is partitioned into two parts: Wired connection using the standard TCP

and wireless connection using a modified version of the TCP. Wired connection is between the fixed host and a middleware station called proxy; while, wireless connection is between the proxy and the mobile hosts. The innovation of this approach is that errors from the wireless connection are corrected at the TCP proxy and not propagated through the fixed network. The drawback of this approach that it disregards TCP's end-to-end semantics and introduces extra overhead as packets are processed twice, one time between the fixed host and the proxy and another time between the proxy and the mobile host. I-TCP architecture is shown in Figure 2.

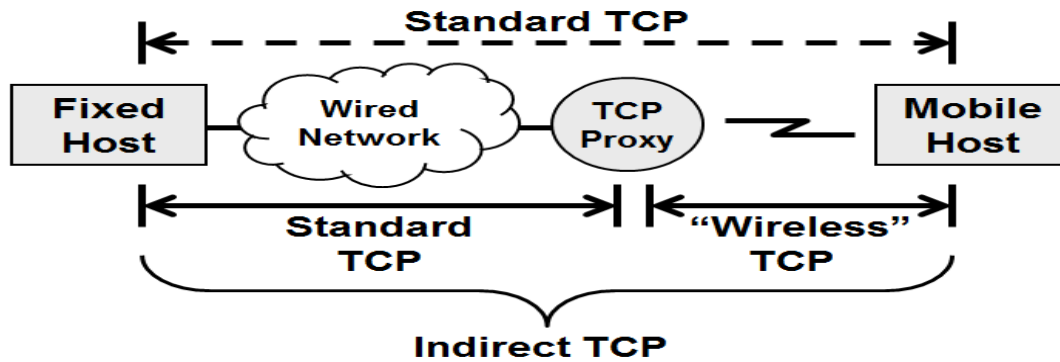


Figure 2. Indirect-TCP Architecture

4.2. Snoop

Snoop [9] is a classic hiding non-congestion loss method in which a Snoop agent is employed between the wired and the wireless network. When the Snoop agent receives a packet from the wired network, it caches it into an internal buffer and then forwards it to the mobile host, and the agent waits for an acknowledgment from the mobile host. When the agent receives an acknowledgment, it checks the status of the packet, if it is inside the cache, then the lost packet is retransmitted to the destination without going back through the wired network; or else, the Snoop agent forwards the acknowledgment to the wired source indicating a congestion problem. The main drawbacks of Snoop are that it is not an end-to-end semantic and it does not completely isolate wireless link errors from the fixed network.

4.3. M-TCP

M-TCP (Multicast TCP) [10] is destined for low bit-rate in wireless links. In this scheme, each and every TCP connection is divided in two segments: The first is a TCP connection from the sender fixed host (FH) to the supervisor host (SH) and it uses the standard TCP protocol. The second connection is between the supervisor host (SH) and the mobile host (MH) and it uses an adapted version of TCP. In action, the FH sends a segment to MH, the SH receives it first then forwards it to the MH which acknowledges it upon receipt. Once SH receives the ACK from the receiver, it forwards it to the FH. In case the MH is disconnected, the SH stops receiving the ACK and considers that the MH has been temporarily disconnected and therefore it sends the ACK to the FH containing the window size of 0. The sender then freezes its timers and sends the backed-off persist packet to

the SH which in turn responds with a zero window size until it receives some non-zero window size. As soon as the window size becomes greater than zero, the SH directly replies with the suitable window size and restarts all its frozen timers. As a result, the sender resumes the transmission at full-speed.

4.4. ECN and Congestion Coherence

Explicit Congestion Notification (ECN) [11] is a network congestion scheme that utilizes two bits in the IP header and two bits in the TCP header to record the status of the network. If congestion occurs, the ECN bits are set to 1 in the last transmitted packet. When the receiver receives the packet with ECN set to 1, it sets the ECN bits of the acknowledgment packet to 1 and sends it to the sender. The sender then reduces its window size to avoid congestion. Congestion coherence (CC) [12] is an improvement over ECN to differentiate deficit. If the ECN bits are set to 1, then the lost packet event is caused by congestion; otherwise, i.e. ECN bits are set to 0, the lost packet event is caused by bit error. Although this scheme can determine the cause of loss exactly, it must be implemented inside the TCP protocol as well as network devices such as routers, switches, and stations. Figure 3 shows an example for an ECN where several packets are discarded due to congestion.

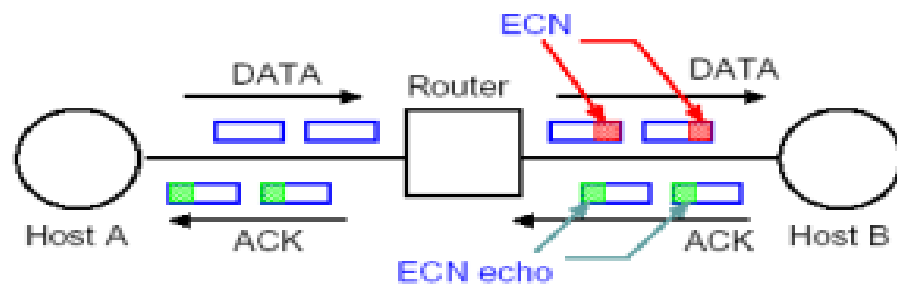


Figure 3. ECN Example

4.5. WTCP

WTCP (Wireless TCP) [13] is an end-to-end semantic mechanism that distinguishes error deficits from congestion deficits by comparing the packet arrival time with the packet departure time. WTCP is a replacement for the TCP protocol which, unlike TCP, does not reduce its transmission rate by half for a packet deficit but uses inter-packet delay as a metric to compute the adaptation rate at the receiver's end; hence, predicting the cause of packet loss and probing the receiver to find out which packet has to be retransmitted. In that sense, WTCP uses rate-based transmission instead of window-based transmission in which the sender does not determine which packet should be retransmitted but uses feedback from the receiver to do so. The major disadvantage of WTCP is that it is independent of the original TCP; and therefore, must be implemented in all the network nodes and the devices.

4.6. Westwood

TCP Westwood [14] distinguishes the reason for packet deficit by estimating the available bottleneck bandwidth. It computes the arrival rate of ACKs and uses the interval of constant feedback ACKs and the packet

size to acquire the optimal network capacity denoted by $SBW[j]$. Additionally, it also computes a smoothed value denoted by $BWE[j]$, by low-pass filtering the sequence of $SBW[j]$.

$$SBW[j] = \text{packet_size} / (\text{current_time} - \text{prev_ACK_time})$$

$$BWE[j] = (1 - t) * (SBW[j] + SBW[j-1]) / 2 + t * BWE[j-1]$$

Where packet_size is the size of the packet, current_time is the most recent time, prev_ACK_time is the time of the last ACK, and 't' is the factor for the low-pass filtering operation. Though Westwood approach tries to approximate the bandwidth of the connection to best find the congestion window size, it sometimes overestimates the available bandwidth making the overall throughput of the wireless network relatively slow.

4.7. TCP Vegas and TCP Veno

TCP Vegas [15] assesses the capacity of the TCP link by discovering the minimum RTT value called $BaseRTT$. The capacity of the backlog queue is computed by the following equations:

$$Expected = \text{cwnd} / BaseRTT, \text{ Actual} = \text{cwnd} / ActualRTT$$

$$Diff = Expected - Actual, \text{ ActualRTT} = BaseRTT + N / Actual$$

$$N = Actual * (ActualRTT - BaseRTT) = Diff * BaseRTT$$

Expected: expected rate of a wireless link, *Actual*: actual rate, *ActualRTT*: real RTT, *cwnd* : current TCP congestion window size, and *N*: size of the queue. The Vegas scheme tries to maintain *N* as small as possible by fine-tuning the TCP window size ahead of time; hence, preventing packet deficit caused by congestion. On the other hand, TCP Veno [16] improves upon Vegas to judge the cause of loss. It sets a threshold, designated by *thres*, to represent the status of the network. If packet deficit occurs and $N > \text{thres}$, then the connection is said to be in bad status and the packet will be considered as congestion loss; otherwise, it will be considered as error deficit. The disadvantages of TCP Vegas/Veno are that it continuously updates the window size leading to a performance overhead. In addition, it can lead to fluctuation in the window size and round trip times; and consequently, causing delay jitter and inefficient bandwidth utilization.

4.8. JTCP

Jitter TCP (JTCP) [17] is a TCP congestion technique to discover packet deficit and to identify whether the deficits are due to congestion or bit error. It is based on the jitter ratio and packet-by-packet delay that are determined by the inter-arrival jitter i.e. the packet spacing at the sender compared with the packet spacing at the receiver for a pair of packets. The inter-arrival jitter can be calculated as follows:

Where 'i' and 'j' designate the index of packet, 'Si' designates sending time for packet 'i', and 'Ri' designates the receiving time for packet 'i'. JTCP can deduce the packet with longer transmitted time and delay it into the router queue until congestion is resolved. JTCP has better performance when compared to other congestion

mechanisms; however, it is required to be implemented in the transport portion of the TCP stack at every endpoint, in addition to network devices, stations, and nodes.

5. Proposed Solution

The authors in this paper propose a TCP congestion control technique suitable for wireless and also for wired networking environments. It is based on the usage of one of the single bit from the reserved bits of the TCP header to determine the type of the link over which a connection is established. If the link is wired, the TCP reserved bit is set to '0' designating a wired mode, else if the link is wireless, the bit is set to '1' designating a wireless mode. In addition, the proposed technique uses the Signal-to-Noise (SNR) ratio to discover the reliability of the link. In wired mode, timeout is considered as a congestion loss; and thus, congestion is avoided by using the classical TCP start-slow algorithm. However, in wireless mode, two scenarios are possible for a packet timeout, both of which are based on SNR ratio: In case SNR is high, i.e., greater than 5dB, that means that the link is reliable and the loss is due to congestion, so the classical TCP congestion mechanism is executed to slow down the burst of packets. In case SNR is low, i.e., less than 5dB, that means that the link is unreliable and the loss is due to error, so the timed-out packet is retransmitted by the sender.

5.1. The Reserved Bits

Transmission Control Protocol (TCP) is one of the essence protocols in the internet protocol suite. It provides reliable and ordered delivery of data stream from an application on one computer to another application on another computer [18]. TCP is part of transport layer that receives data stream from the application layer for processing. Then the data stream is segmented into small data units called packets, and appended to a TCP header creating a TCP segment. A TCP segment consists of header and data section. The header section consists of ten compulsory fields, and an optional extension field; while, the data section trails the header and carries the payload data for the application.

The TCP header consists of several fields; the source port, destination port, sequence number, data offset, flags, and checksum are few to mention. One overlooked field is the reserved field which is composed of three bits that are reserved by the original TCP design for future use. In modern TCP implementations, these bits are never used and are set to zero. The proposed technique exploits this reserved field to discover the type of the link over which the connection is established. In fact, only the first bit is used that is set to '0' to indicate a wired communication or set to '1' to indicate a wireless communication. This mechanism must be implemented in the TCP stack of every network device including operating system, switches, routers, and stations so that packets are leveraged by the proposed TCP congestion technique. Figure 4 depicts the original TCP header format along with the reserved bits leveraged by the proposed TCP congestion technique.

5.2. The Congestion Algorithm

In digital communication, a signal-to-noise ratio (SNR) is a measure used to compare the level of a desired signal to the level of background noise [19]. Higher limit of SNR is considered and better. The lower limit of SNR is usually considered to be around 5dB, while 12dB is fairly sufficient for most conditions. In practice, the

SNR ratio is detected at the initial stage of communication session setup. Generally, low SNR leads to high bit error rate and CRC error rate causing packet deficit and timeouts. Mathematically, SNR is defined as the ratio of signal power to the noise power, and it can be computed with the help of the below equation:

$$\text{SNR} = \frac{P_{\text{signal}}}{P_{\text{noise}}}$$

SNR is expressed in decibel units according to the following equation:

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \left(\frac{P_{\text{signal}}}{P_{\text{noise}}} \right) = P_{\text{signal,dB}} - P_{\text{noise,dB}}$$

When this proposed technique is operating in wireless mode, it exploits the SNR ratio of the communication line to decide whether the timed-out packet was due to congestion or error deficit. Once the TCP connection is established and the communication begins, the congestion algorithm initializes the congestion window '*cwnd*' to one segment. Then, the first bit of the reserved field is set according to the link type, i.e., $b=0$ for wired and $b=1$ for wireless connection. Packets are then sent to the receiver. When they are successfully acknowledged, the congestion window *cwnd* is incremented by one segment, making its size to two segments, then to four segments, then to eight segments, and so on doubling the window size each time until the size advertised by the receiver is reached or until congestion occurs. When congestion occurs, packets may be lost, which triggers a timeout condition at the sender. In this situation, the sender immediately checks the reserved bit '*b*'. If it is equal to '0' (wired link), then timeout is considered to be due to congestion. The size of the congestion window '*cwnd*' would be set to one half of the current size and the sender resumes the packet bursts. In contrast, if the reserved bit '*b*' is equal to '1' (wireless link), the SNR ratio of the connection is checked and if it is within a high range, i.e., greater than 5dB ($\text{SNR} > 5\text{dB}$), then timeout is considered to be due to congestion and the window size is reduced by half by one segment and next packet is transmitted. However, if SNR ratio is within a low range, i.e., less than 5dB ($\text{SNR} < 5\text{dB}$), then timeout is considered to be due to error and the timed-out packet is retransmitted to the receiver. The flowchart of the proposed congestion algorithm is illustrated in Figure 5.

The pseudo-code of the proposed algorithm is as follows:

1. Initialize the congestion window *cwnd* to one segment.
2. Start transmitting packets to receiver.
3. For each acknowledgement received, the congestion window '*cwnd*' is incremented by one segment.
4. If timeout occurs for particular packet '*p*', then check the value of the reserved bit '*b*' for packet *p* : If $b=0$ (wired link), then reduce *cwnd* to half of its previous size (indicating a congestion) or else if $b=1$ (wireless link),
5. Then check the SNR ratio of the link : If $\text{SNR} > 5\text{dB}$, then reduce *cwnd* to half of its previous size (indicating a

congestion) or else if $SNR < 5dB$, then no change in the current size of $cwnd$, but retransmit packet p (indicating an error)

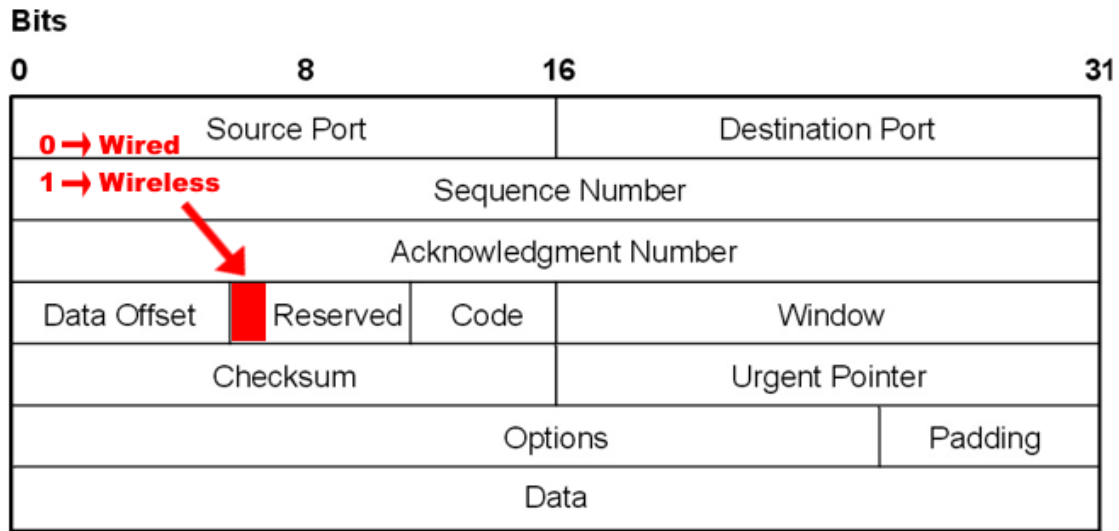


Figure 4. TCP Header with Reserved Bits . Leveraged by the proposed TCP congestion scheme.

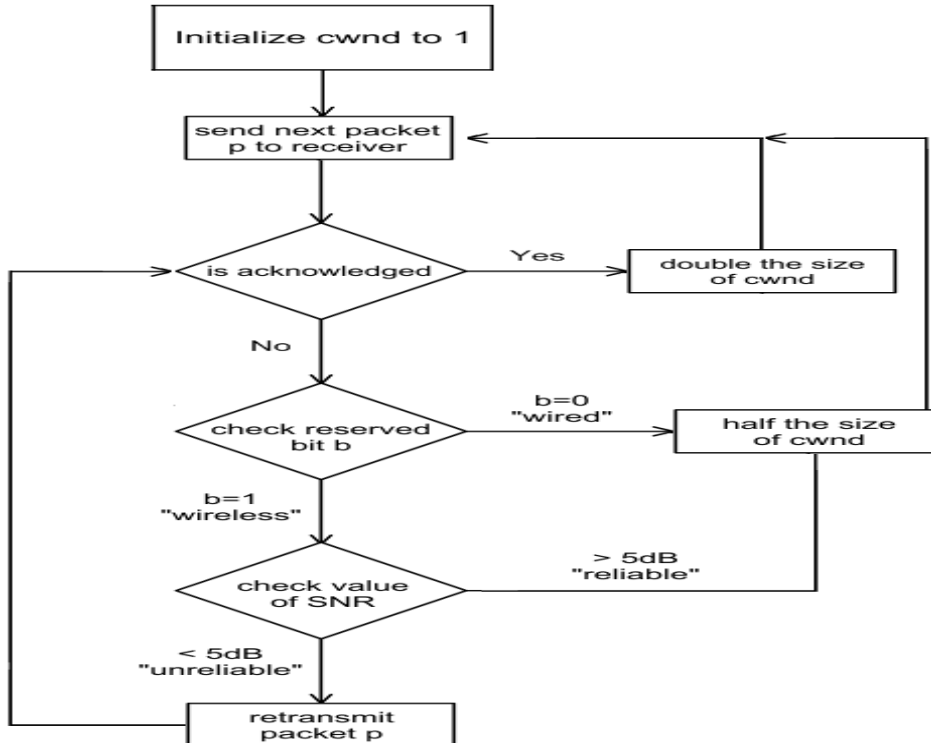


Figure 5. Flowchart of the Proposed Congestion Algorithm

5.3. Advantages of the proposed technique

End-to-end semantics: This new proposed technique maintains true end-to-end semantics, without the involvement of intermediate nodes between the sender and the receiver.

Performance: It maintains high level of performance especially that no intermediate nodes are involved which eliminates any extra processing overhead and the need for extra buffer space.

No change in code: It manages the source code of the sender and the receiver without the need to alter and recompile the communicating applications.

Compatibility: It provides compatibility by maintaining the original TCP structure with one exception is the integration of the proposed congestion algorithm into the standard TCP RFC 793 stack so as to take advantage of the TCP reserved field and the SNR ratio of the underlying transmission link.

6. Simulation & Results

The proposed mechanism and the standard TCP (RFC 793) [20] were simulated using Network Simulator 3 (NS-3). For simulation purpose, a network model has been constructed. It is a wireless topology network in which 'C' is a client node, 'R1' is a router, and 'R' is a receiver node. The bandwidth between 'C' and 'R1' is 100Mbps with 5ms propagation time; while, bandwidth between R1 and R is 80Mbps with 2ms propagation time. The packet size is 1 KB and the size of queue is 100. Figure 6 illustrates the simulated wireless network model.

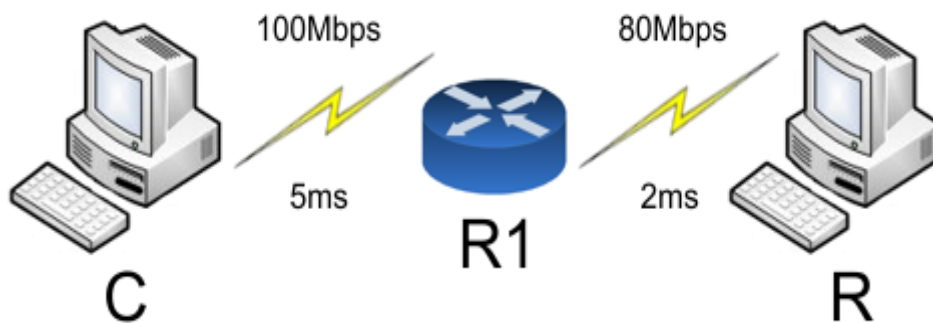


Figure 6. Wireless Network Model

The simulation was run for 350 seconds and two aspects were evaluated in presence of packet deficit which is caused by error: The network throughput and the congestion window size. Figure 7 illustrates the simulation results for network throughput of the proposed mechanism and the standard TCP protocol (RFC 793).

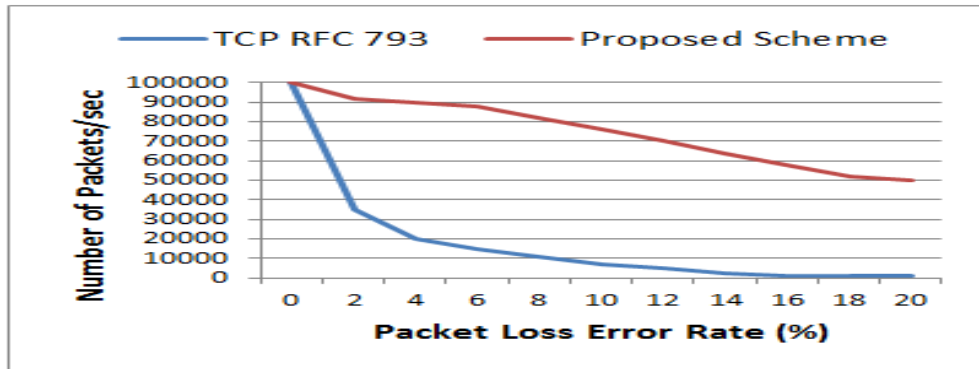


Figure 7. Simulation Results for Network Throughput

In the above results, the throughputs of both mechanisms are the same for 0% packet loss rate. However, when the packet loss rate increased, the deviation became more visible. In fact, the throughput of the TCP RFC 793 dropped exponentially with the increase of packet deficit; while, the proposed mechanism retained a more stable throughput throughout the simulation. This can be explained by the fact that TCP RFC 793 reduces its window size by half, when a timeout occurs (unacknowledged packet due to congestion loss), leading to a decrease in the network throughput. On the other hand, the proposed mechanism can detect the rate of packet loss by finding the SNR ratio of the link and consequently decide whether to retransmit the packet or reduce the connection bandwidth by half. In the simulation, the packet deficit was detected due to error and not due to congestion. The greater the packet loss lowers the SNR ratio. Therefore, the proposed mechanism retransmits timed out packets instead of decreasing the window size and the connection bandwidth.

Furthermore, the simulation was executed for another 350 seconds to compute the congestion window size in presence of packet loss. Figure 8 and 9 represents the simulation results for congestion window size of the proposed mechanism and the standard TCP protocol (RFC 793).

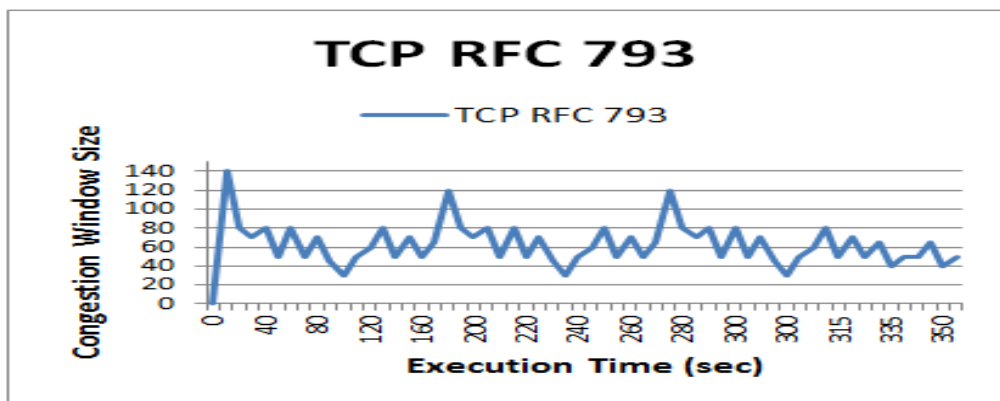


Figure 8. Simulation Results for Window Size of TCP RFC 793

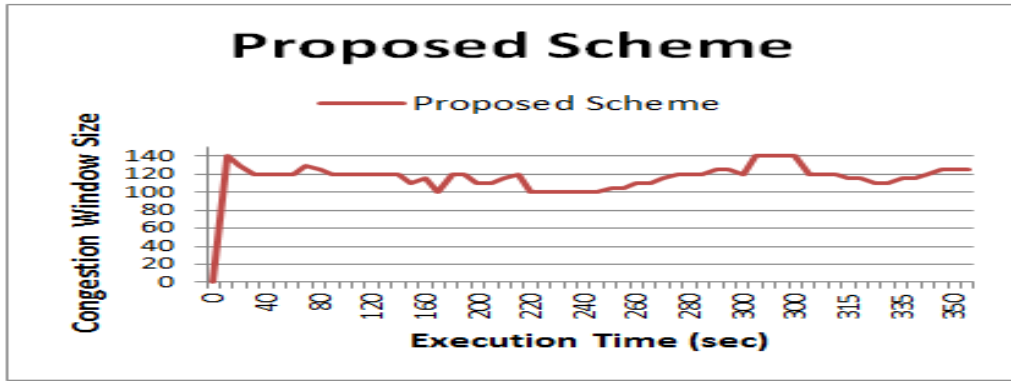


Figure 9. Simulation Results for Window Size of the proposed mechanism

According to the above results, with the increase in execution time, there was increase in packet loss which led the TCP RFC 793 to consider the situation as congestion and as started reducing the congestion window size. Initially it was started at 140 bytes then it got reduced up to nearly to 20 bytes. The consequences of this reduction are a lower throughput and a decrease in the connection speed. In contrast, as the proposed scheme can differentiate between packet loss due to congestion and the one due to error, most of the window size was maintained with little fluctuation which ranged between 140 and 100 bytes. This fluctuation in the window size was due to timed-out packets in presence of a high SNR ratio ($SNR > 5dB$), which indicated zero errors but network congestion. Nevertheless, the window size was maintained in the presence of timeouts that were due to low SNR ratio ($SNR < 5dB$), indicating no congestion but network errors.

7. Conclusions & Future Work

In this paper the authors presented a novel strategy for solving the TCP performance problem for wireless networks. Its objective is to allow the TCP protocol to distinguish between transmission time outs due to congestion and the transmission timeouts due to errors. This new strategy uses the TCP reserved field to identify the network type over which communication occurs, in addition to the SNR ratio to determine the reliability of the link so that better decision can be made regarding whether to reduce packet burst or retransmit a timed-out packet. Simulation of the proposed scheme clearly showed that it successfully implemented and managed to determine the cause of packet deficit in wireless networks and take the right decision in situations where timeouts were due to error and not to congestion, that is, retransmitting the timed-out packet instead of reducing the congestion window size and consequently not diminishing the burst of packets and the overall throughput of the network.

As future work, Cyclic Redundancy Check (CRC) and Header Error Check (HEC) metrics can be added as additional parameters to the proposed mechanism so that the reason for transmission timeouts in wireless networks can be predicted and determined in an efficient manner.

References

- [1] R. Stewart, C. Metz, "SCTP: new transport protocol for TCP/IP", IEEE Internet Computing, vol. 5, no. 6, pp. 64-69, 2001.
- [2] Van Jacobson, and Michael J. Karels, "Congestion Avoidance and Control", Proceedings of ACM SIGCOMM '88, pp: 314-329, 1988.
- [3] Larry L. Peterson, Bruce S. Davie, Computer Networks, A Systems Approach, 5th ed, Morgan Kaufmann, 2011.
- [4] G. Xylomenos, G.C. Polyzos, P. Mahonen, and M. Saaranen, "TCP Performance Issues over Wireless Links", IEEE Communications Magazine, 2001.
- [5] M. Allman, V. Paxson, and W. Stevens, "TCP Congestion Control", RFC 2581, 1999.
- [6] Tanenbaum, Wetherall, Computer Networks, 5th ed, Prentice Hall, 2010.
- [7] Ye Tian, Kai Xu, and Ansari N, "TCP in Wireless Environments: Problems and Solutions", IEEE Communications Magazine, vol. 43, no. 3, pp. 27-32, 2005.
- [8] Bakre, Badrinath, "I-TCP: Indirect TCP for Mobile Hosts", In Proceedings of ICDCS 95, 1995.
- [9] Hari Balakrishnan, Srinivasan Seshan, Elan Amir, and Randy H. Katz, "Improving TCP/IP Performance over Wireless Networks", Proceedings of the 1st ACM Conference on Mobile Computing and Networking, Berkeley, CA, November 1995.
- [10] K. Brown and S. Singh, "M-TCP: TCP for Mobile Cellular Networks", ACM Computer Communications Review, vol. 27, no. 5, pp. 19-43, 1997.
- [11] Ramakrishnan and Floyd, "A Proposal to add Explicit Congestion Notification (ECN) to IP", Internet Draft, January 1999.
- [12] Chunlei Liu, and Jain R., "Approaches of Wireless TCP Enhancement and A New Proposal Based on Congestion Coherence", System Sciences, 2003.
- [13] P. Sinha, N. Venkitaraman, R. Sivakumar, and V. Bharghavan, "WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks", ACM Mobicom, Seattle, WA, 1999.
- [14] Saverio Mascolo, Claudio Casetti, Mario Gerla, M. Y. Sanadidi, and Ren Wang, "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links", ACM Mobicom, 2001.

- [15] Lawrence S. Brakmo, and Larry L. Peterson., “TCP Vegas: End to End Congestion Avoidance on a Global Internet”, IEEE Journal on selected areas in communications, vol. 13, no. 8, 1995.

- [16] Cheng Peng Fu, and Soung C. Liew, “TCP Veno: TCP Enhancement for Transmission over Wireless Access Networks”, IEEE Journal on selected areas in communications, vol. 21, no. 2, 2003.

- [17] Wu E.H.-K., and Mei-Zhen Chen, “JTCP: Jitter-based TCP for Heterogeneous Wireless Networks”, Selected Areas in Communications, IEEE Journal, vol. 22, no. 4, pp. 757-766, 2004.

- [18] G. R. Wright, and W. R. Stevens, TCP/IP Illustrated, Volume 2 (The Implementation), Addison Wesley, 1995.

- [19] Rafael C. González, Richard Eugene Woods, Digital Image Processing, Prentice Hall, 2008.

- [20] J. Postel, Transmission Control Protocol, RFC 793, September 1981.