A Survey on TCP Westwood
S.Thilagavathi, J.Vanitha

Abstract—TCP Westwood (TCPW in short) is a sender side modification of the TCP congestion window algorithm. An important distinguishing feature of TCP Westwood with respect to previous wireless TCP “extensions” is that it does not require inspection and/or interception of TCP packets at intermediate (proxy) nodes. TCP Westwood introduces a mechanism of faster recovery to avoid overly conservative reduction of the congestion window after a congestion episode by taking into account the end-to-end estimation of available bandwidth. The estimated bandwidth is used to compute congestion window and slow start threshold settings after a congestion episode that is after three duplicate acknowledgements or a timeout. The increase after congestion is Additive Increase Additive Decrease in TCPW compared to AIMD (Additive Increase Multiplicative Decrease) of Reno. TCPW is extremely effective in wired and wireless networks. TCPW does not require any support from lower layers, and thus strictly adheres to layer separation and modularity principles. Also, TCPW does not require that any TCP option be used in segment headers. This paper use a theoretical approach based on the use of various algorithms in TCPW and discuss about their performance in different wireless networks.

Index Terms—Congestion window, Faster recovery, Slow Start threshold, TCP Westwood, Wireless networks.

I. INTRODUCTION

TCP and IP were inspired and originally drafted in RFC 793 and RFC 791, respectively, in September 1981. The Transmission Control Protocol (TCP) is a reliable, connection-oriented, full-duplex, byte-stream, transport layer protocol. It is an end-to-end protocol that supports flow and congestion control, and is used by many end-user applications. In fact, the vast majority of today’s Internet traffic uses TCP.

TCP was designed for wired networks and has been highly tuned over the years. Although TCP is very efficient on wired networks, it has been shown to perform poorly on wireless. The main reason for this poor performance is that the TCP congestion control mechanism cannot distinguish between packet losses occurring randomly in wireless channels and those due to network congestion. Therefore, the TCP congestion control mechanism reduces, even when not necessary, the transmission rate.

TCP Reno [2] is one of the most widely adopted TCP schemes. TCP maintains two state variables to regulate the transmission rate: congestion window (cwnd) and slow start threshold (ssthresh). The latter sets the cwnd value that discriminates between the slow-start and congestion avoidance phases. At the beginning of the connection, the source increases cwnd exponentially (slow start) until the network drops packets and a condition of congestion is recognized. In response to this, TCP Reno sets ssthresh to one half of the bytes in flight. When cwnd reaches the new ssthresh value, TCP enters a congestion avoidance phase during which cwnd is increased linearly. This scheme assumes that ssthresh gives an estimate of the available bandwidth, and uses the congestion avoidance phase to probe gently for extra bandwidth.

However, the bandwidth estimate is accurate only if the first packet loss occurs when the sending rate has reached the available rate. If the loss is due to transmission error, as in wireless channels, ssthresh can be set erroneously to a small value, thus limiting the sending rate and degrading the throughput performance. The existing versions of TCP, like Reno or NewReno, experience heavy throughput degradation over channels with high error rate, such as wireless channels.

TCP/IP has become the dominant communication protocol suite in today’s multimedia applications. A large amount of internet traffic is carried by TCP. TCP/IP needs to depart from its original wired network oriented design and evolve to meet the challenges introduced by the wireless portion of the network.

To avoid such limitation and degradation, several schemes have been proposed and are classified as end-to-end protocols, where loss recovery is performed by the sender, split connection protocols, that break the end-to-end connection into two parts at the base station, and link layer protocols based on a combination of ARQ and FEC.
techniques. The link-layer schemes have been shown to improve significantly the performance of TCP sources when transmitting over wireless links. However, end-to-end techniques, even if not as effective as link-layer protocols, can achieve further gain in performance by using more sophisticated bandwidth estimation algorithms in the TCP congestion control scheme.

In the end-to-end approach, the ability to accurately probe for the available bandwidth is the key to better performance, which is still a great challenge. Examples of this approach include TCP Reno employs reactive flow control, TCP SACK is a selective ACK (SACK) option for TCP, TCP Vegas estimates the backlogged packets in the buffer of the bottleneck, TCP Veno adopts the same methodology as Vegas, TCP Westwood is a rate based end to end approach, TCP-Jersey adapts the sending rate proactively according to the network condition.

The aim of this paper is to investigate the performance of TCP Westwood with various algorithms in the different wireless networks. The paper is organized as follows. Section 2 provides an overview of TCP Westwood, Section 3 provides TCPW with various algorithms and section 4 concludes the paper.

II. TCP WESTWOOD OVERVIEW

TCP Westwood (proactive approach)[2] is a rate based end-to-end approach in which the sender estimates the available network bandwidth dynamically by measuring and averaging the rate of returning ACKs. It calculates the explicit available bandwidth and uses it to guide the sending rate. This estimate is used to set the ssthresh and the cwnd after congestion events, such as the receipt of three duplicate ACKs or coarse timeout expirations. This recovery mechanism avoids the blind halving of the sending rate of TCP Reno after packet losses and enables TCP Westwood to achieve a high link utilization in the presence of the random, sporadic loss typical of wireless links. TCP Westwood claims improved performance over TCP Reno and SACK while achieving fairness and friendliness. The end-to-end approach maintains the network layer structure and requires minimum modifications at end hosts and in some cases also the routers.

Algorithm WESTWOOD 1:
It proposes a simple scheme for the TCP source to estimate the available bandwidth and use the bandwidth estimation to recover faster, thus achieving higher throughput.

In this algorithm, sequence of bandwidth samples \( \text{sample}_BWE[k] \) is obtained using the ACK arrivals and evaluates a smoothed value, \( BWE[k] \), by low-pass filtering the sequence of samples, as described by the following pseudo code:

```plaintext
if (ACK is received)
    \( \text{sample}_BWE[k] = \frac{\text{acked} \times \text{pkt_size} \times 8}{\text{now - last_ACK_time}}; \)
    \( BWE[k] = (1 - \beta) \times (\text{sample}_BWE[k] + \text{sample}_BWE[k-1]) / 2 + \beta \times BWE[k-1]; \)
endif
```

where \( \text{acked} \) is the number of segments acknowledged by the last ACK, \( \text{pkt_size} \) is the segment size in bytes, \( \text{now} \) is the current time, \( \text{last_ACK_time} \) is the time the previous ACK was received, \( k \) and \( k-1 \) indexes indicate the current and previous value of the variables, and \( \beta \) is the pole used for the filtering (in [17] a value of \( \beta = \frac{1}{2} \) is suggested)

Algorithm WESTWOOD 2:
The improved version of the TCP Westwood algorithm described in the following pseudo code adopts a nonlinear filtering technique:

```plaintext
if (ACK is received)
    \( \text{ACK_interval} = \text{now - last_ACK_time}; \)
    \( \text{sample}_BWE[k] = \frac{\text{acked} \times \text{pkt_size} \times 8}{\text{ACK_interval}}; \)
    \( \text{pole} = \frac{2\times\text{tau} - \text{ACK_interval}}{2\times\text{tau} + \text{ACK_interval}}; \)
    \( BWE[k] = (1 - \text{pole}) \times (\text{sample}_BWE[k] + \text{sample}_BWE[k-1]) / 2 + \text{pole} \times BWE[k-1]; \)
Endif
```
**Algorithm WESTWOOD 3:**
The estimation algorithm of TCP Westwood was modified further, the new approach adopting time varying coefficients in the filter with both adaptive gain and adaptive sampling. The following pseudo code specifies the modified approach:

```plaintext
sample_BWE[k]= (Bytes received in T[k])/T[k]
pole[k]=(2*tau[k] - ACK_interval)/(2*tau[k] + ACK_interval);
BWE[k]= pole[k]*BWE[k-1] + (1 - pole[k])* sample_BWE[k];
```

where \( \tau[k] \) (the parameter that determines filter gain) adapts to path conditions, as does the bandwidth sample \( \text{sample}_BWE[k] \), calculated over a time interval \( T[k] \).

TCP Westwood has brought many significant improvements and research aspects into the world. Its renowned performance has been proved via simulation and real testbed results for both wired and wireless networks.

### III. TCP WITH VARIOUS ALGORITHMS

**A. TCPW with Bulk Repeat (TCPW BR)**
TCPW with Bulk Repeat (TCPW BR)\(^5\) proposed to deal with high bandwidth, error-prone wireless links, a configuration which will become pervasive in Next Generation wireless networks. BR has only three sender side modifications to TCP, (1) Bulk Retransmission: retransmitting all unacknowledged packets immediately when more than one loss is detected in a window; (2) Fixed Retransmission Timeout: using a fixed timeout value instead of exponential backoff when consecutive losses occur; and (3) Intelligent Window Adjustment: keeping \( cwnd \) fixed in spite of losses.

The sender enables the BR modifications only when it suspects the losses to be caused by errors rather than congestion. A Loss Discrimination Algorithm (LDA) is used to distinguish between congestion and errors. TCPW BR can solve the TCP degradation problem caused by high loss rates.

**B. TCPW ABSE**
TCPW ABSE (Adaptive bandwidth share Estimation)\(^3\) improves throughput in a mobile environment. This is the first step in the study of TCPW when used in adhoc networks. TCPW ABSE is a sender side modification of TCP NewReno. The ABSE algorithm adapts to the congestion level in performing its bandwidth sampling and employs a filter that adapts to the round trip time and to the rate of change of network conditions. The bandwidth share estimation is computed using a time varying coefficient, exponentially-weighted moving average (EMWA) filter, which has both adaptive gain and adaptive sampling.

**Experiments description:**

**Scenario 1:** Indoor Environment with a static wireless station
TCPW ABSE doesn’t perform better than TCP NewReno. This is because it is likely to be used in packet loss environment.

**Scenario 2:** Wireless station moves in an indoor environment and hide wireless card.
The behavior of TCPW ABSE when the wireless station moves continuously achieves a better rate than that of TCP NewReno.

**Scenario 3:** Mixture of indoor and outdoor environments.
TCPW ABSE achieves better average throughput than TCP New Reno.

**C. TCPW-A (TCP Westwood with agile probing)**
TCPW-A, (TCP Westwood with agile probing), a sender-side only enhancement of TCPW, that deals well with highly dynamic bandwidth, large propagation times and bandwidth, and random loss in the current and future heterogeneous network. TCPW-A achieves this goal by incorporating the following two mechanisms into basic TCPW algorithm: The first mechanism is Agile probing, which is invoked at connection start-up and after extra available bandwidth is detected. Agile probing adaptively and repeatedly resets ssthresh based on ERE (Eligible Rate Estimate).

Each time the ssthresh is reset to a value higher than the current one, cwnd climbs exponentially to the new value. This way, the sender is able to grow cwnd efficiently to the maximum value allowed by current conditions without overflowing the bottleneck buffer with multiple losses. The second mechanism load gauge mechanism
which identifies the availability of persistent extra bandwidth in congestion avoidance, and invokes agile probing accordingly. TCPW-A only requires sender-side modification, thus much easier to deploy.

**D. TCPW with RED and ECN**

AQM[7] aims to improve overall network bandwidth utilization, smooth buffer occupancy in routers, and provide better fairness among transport layer connections. The earliest and most prominent of the AQM schemes is RED. In RED, packets are randomly dropped from the router buffer when the occupancy of the buffer exceeds certain thresholds. In Explicit Congestion Notification (ECN), routers notify a sender explicitly of congestion, instead of the implicit notification resulting from a packet drop. Because of the explicit congestion notification in ECN, the sender ability in discriminating the cause of packet losses is indeed enhanced.

TCPW and RED interoperate well and exhibits better fairness in cases of connection with asymmetric propagation time, as well as for symmetric connections. TCPW benefits from ECN as packet delays are significantly reduced and making better estimate of eligible rate rate estimation.

**E. TCPW in Multiple-Path Cases**

When multiple paths [4] are deployed for fault tolerance, load balancing, security, etc., generic TCP variants do not survive the multiple path environments. This is primarily because TCP blindly treats duplicate ACKs as indications of packet losses and incipient network congestion even though their corresponding data packets have been correctly (but out-of-sequence) delivered to their destination. This limitation is avoided by the bandwidth estimating feature of TCP Westwood with which the traffic source can estimate bandwidth by intelligently monitoring ACKs and adjusting ssthresh and cwnd more adequately to represent the current network condition. This new congestion control mechanism outperforms the “blind” approach of conventional TCPs to handle threshold and window. Consequently, TCP Westwood is deemed to be the preferred approach for an Internet featuring multiple paths for load balancing and for fast response fault tolerant QoS provisioning.

**IV. CONCLUSION**

This paper gives an introduction of TCPWESTWOOD and its types. TCP Westwood controls the window using end-to-end connection bandwidth share estimation. And attempts to make a more “informed decision”. In this way TCPW ensures both faster recovery and more effective congestion avoidance. This paper also tries to survey the performance of TCPWESTWOOD with other algorithms in various wireless networks, limiting to theoretical review of the existing papers. The study reveal the benefits of the TCPW: TCPW exhibits significant improvement in various environments compared to other TCP variants.

**REFERENCES**


AUTHOR BIOGRAPHY

Ms.S.Thilagavathi is an Assistant Professor in the department of Electronics and Communication Engineering, VV College of Engineering, Tirunelveli. She received her M.E. from Anna University, Chennai. She is having 10 years of experience and published 4 papers in National Conferences and 1 paper in international journal.

Ms.J.Vanitha is an Assistant Professor in the department of Electronics and Communication Engineering, VV College of Engineering, Tirunelveli. She received her M.E. from Anna University, Chennai. She is having 8 years experience and published 2 papers in National Conference, 3 papers in International Conference and 1 paper in international journal.