CONGESTION CONTROL WITH DYNAMIC THRESHOLD ADAPTATION AND CROSS-LAYER RESPONSE FOR TCP OVER IEEE 80.11 WIRELESS NETWORKS

Rung-Shiang Cheng[1], Der-Jiunn Deng[2] and Han-Chieh Chao[3]
[1]*Department of Computer and Communication, Kun Shan University, Yung Kang, Taiwan*
[2]*Department of Computer Science and Information Engineering, National Changhua University of Education Changhua City, Taiwan*
[3]*Department of Electronic Engineering, National Ilan University, Ilan City, Taiwan*

Keywords: TCP, Congestion control, IEEE 802.11, Cross-layer response.

Abstract: Wireless technologies provide mobile access and enable rapid and cost-effective network deployment. But a wireless link is generally accompanied by high interference, transmission errors and a varying latency. The erratic packet losses usually lead to a curbing of the flow of segments on the TCP connection, and thus limit TCP performance. This paper presents a threshold control mechanism with cross-layer response approach for improving TCP Vegas performance in IEEE 802.11 wireless networks. By making slight modifications to the legacy IEEE 802.11 MAC and TCP, the numerical results reveal that the proposed scheme provides a significant improvement in TCP performance under IEEE 802.11 wireless environments.

1 INTRODUCTION

The proliferation of mobile computing devices in recent years has led to the extensive deployment of wireless communication networks. More and more devices (such as PCs, Laptops, Smart phones, Tablets) are equipped with network interface cards that integrate wireless technologies (such as 3G/UMTS/GPRS and Wi-Fi) to connect to the networks. New standards for wireless communications (IEEE Standard 802.11, 1999) (IEEE Standard 802.16-2004, 2004) have greatly improved data transmission rate and prompted the development of high-speed wireless communication services capable of supporting the anticipated introduction of a wide range of new applications over the coming years.

Recently, TCP and IEEE 802.11 have become the most commonly applied transport protocol for Internet applications and the standard of choice for wireless local area networks, respectively. In IEEE 802.11, wireless communication applications are commonly constrained by scalability problems. which fail to achieve a reasonable throughput as the number of hosts increases since CSMA/CA access schemes have extremely low spatial-reuse efficiency. As a result, the link layer reliability deteriorates rapidly as the scale of the network increases, and thus the number of link layer contentions increases significantly.

TCP Reno (Paxson, 1999) is currently the most widely applied transport layer protocol for Internet congestion control. Although Reno-based TCP (e.g. NewReno, SACK) (Floyd, 1999) (Mathis, 1996) is the most commonly used TCP, however, its aggressive behavior causes a severe oscillation of the congestion window size, and the resulting periodic congestion leads to a limited throughput. Hence, several bandwidth estimation schemes for TCP have been proposed (Jain, 2002). Some TCP bandwidth estimation schemes utilize the end-to-end delay information derived from the measured packet round-trip time (RTT). The delay-based approach, originally described by Jain (Jain, 1989), is best represented by TCP Vegas (Brakmo, 1995).

Unlike Reno-based TCP, TCP Vegas implements a more precise congestion avoidance mechanism based on packet delay rather than the packet loss. Although TCP Vegas yields both a higher throughput and a lower packet-loss ratio than TCP Reno does, however, TCP Vegas fails to obtain a fair share of the bandwidth when competing with TCP Reno in heterogeneous networks. Therefore, in (Cheng, 2010), we proposed an adaptive TCP

Cheng R., Deng D. and Chao H..
CONGESTION CONTROL WITH DYNAMIC THRESHOLD ADAPTATION AND CROSS-LAYER RESPONSE FOR TCP OVER IEEE 80.11 WIRELESS NETWORKS.
DOI: 10.5220/0003677100950100
In *Proceedings of the International Conference on Wireless Information Networks and Systems* (WINSYS-2011), pages 95-100
ISBN: 978-989-8425-73-7

congestion control scheme which dynamically adjusts the threshold parameters being used to define the congestion window size in such a way as to improve the buffer occupancy of TCP Vegas when competing with TCP Reno.

Since the performance of data transfer not only relies on the support of the end-to-end transport layer, but also on the quality of the hop-by-hop communication link, the present study attempts to enhance the legacy IEEE 802.11 MAC and TCP protocols such that TCP is rendered capable of differentiating between corruption and congestion losses by using information received from the MAC layer. In this paper, the current study proposes a collaborative cross-layer approach based on the threshold adjustment mechanism and negative-ACK scheme for enhancing the performance of TCP Vegas in wireless networks.

The remainder of this paper is organized as follows. Section 2 briefly describes the IEEE 802.11 media access control scheme and the TCP protocol. Section 3 describes the implementation of the cross-layer TCP Vegas and IEEE 802.11 MAC protocol. Section 4 compares the performance of the cross-layer scheme with that of TCP Vegas by performing a series of numerical simulations. Finally, Section 5 presents some brief conclusions.

2 RELATED WORKS

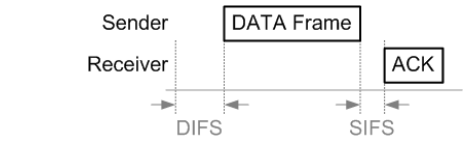
2.1 Media Access in IEEE 802.11 MAC

The IEEE 802.11 standard defines two types of service, namely a contention-free polling-based point coordination function (PCF) and a contention-based distributed coordination function (DCF). In infrastructure-based networks, DCF can operate either alone or in conjunction with PCF. However, in ad hoc networks, DCF operates alone. DCF is the basic access method for the 802.11 standard and is based on the conventional carrier sense multiple access with collision avoidance (CSMA/CA) scheme. DCF comprises both a basic access method and an optional channel access method based on RTS/CTS exchanges.

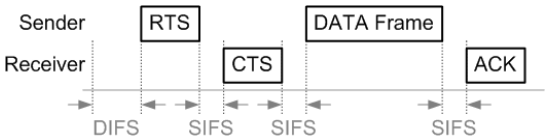
In 802.11, priority access to the wireless medium is controlled by applying an inter-frame space (IFS) time between the frame transmissions. To prevent collisions, the transmitter is obliged to wait for the availability of the channel for a specified interval of time, designated as the distributed inter-frame space (DIFS) before sending a frame. If the medium is currently busy or becomes busy during this interval, the transmitter defers the frame transmission until it detects a DIFS.

Due to the half-duplex nature of wireless interfaces, stations in the network are unable to detect a collision simply by listening to their own transmissions. Therefore, an immediate positive acknowledgment technique is employed to confirm the successful reception of a frame. Specifically, having received a frame, the receiver waits for a short inter-frame space (SIFS) and then transmits a positive MAC acknowledgment to the transmitter confirming that the frame has been correctly received. The SIFS is deliberately assigned a shorter interval than the DIFS in order to assign the receiving station a higher priority than any other stations waiting for making a transmission. The ACK is only transmitted when the frame is received correctly, and hence if the transmitter does not receive an ACK, it assumes that the data frame must have been lost and therefore schedules a retransmission. Figure 1(a) illustrates the basic operations involved in 802.11 DCF.

To alleviate the hidden-station problem (Kim, 1997), 802.11 DCF also provides an optional channel access method using a virtual carrier sensing mechanism based on two special control frames, namely request-to-send (RTS) and clear-to-send (CTS). As shown in Figure 1(b), before transmitting a frame, the transmitter transmits an RTS frame asking the receiver if the medium in its vicinity is free. Once the receiver receives this RTS frame and assumes that no interfering transmission is present, it waits for the specified SIFS interval and then sends a CTS frame to the transmitter. Both transmitter and receiver neighbors overhear these frames and consider the medium reserved for the transmission duration.



(a) Basic access mechanism.



(b) Virtual carrier sensing mechanism.

Figure 1: Media access mechanisms in IEEE 802.11 DCF.

The IEEE 802.11 MAC layer provides a reliable communication link by handling packet delivery using the MAC positive ACK and retransmission

mechanisms (i.e. an automatic repeat request, (ARQ)). However, if several transmission attempts of a packet (e.g. 7 times for the basic access mechanism and 4 times for the virtual carrier sensing mechanism) by a sender in a wireless network fail, the sender simply assumes that the receiver is no longer within its transmission range and gives up its transmission attempt.

As described above, the stop-and-wait schemes such as the IEEE 802.11 ARQ scheme are highly inefficient due to the idle time spent by the sender waiting for the receiver's positive ACK message to arrive. When the end-to-end path comprises multiple wireless links, the accumulated backoff delays and losses further reduce the network throughput, and thus increasing the underutilization of the wireless links. Hence, it is desirable to enhance the interaction between the MAC layer and the transport layer in order to achieve the level of transmission throughput demanded by communication protocols such as TCP.

2.2 Congestion Control in TCP Vegas

TCP Vegas (Brakmo, 1995) adopts the measured RTT to calculate the number of data packets which the sender can inject into the network without incurring packet loss. While doing this, Vegas attempts to maintain the number of the packets queued in the router buffer within two limits, denoted as α and β, respectively. If the measured RTT increases, Vegas recognizes that the network is becoming congested and throttles the congestion window down. Conversely, if the RTT reduces, Vegas recognizes the availability of more bandwidth and therefore increases the congestion window size accordingly. Each time a new ACK is received, the sender updates the congestion window size in accordance with the difference (δ) between the expected throughput and the actual throughput. According to the definition in (Chengrs, 2010), δ is represented as the number of the packets queued in the buffer of end-to-end path and can be expressed as

$$\begin{aligned}\delta &= (\text{Expected throughput} - \text{Actual throughput}) \times minRTT \\ &= \left(\frac{W}{minRTT} - \frac{W}{measuredRTT}\right) \times minRTT \end{aligned} \quad (1)$$

Here, the expected throughput represents the available bandwidth for the current TCP connection in the absence of network congestion, while the actual throughput represents the bandwidth currently used by the TCP connection. Finally, minRTT represents the minimum value of the measured RTT.

Vegas utilizes two threshold parameters α and β to control the maximum difference between the expected and actual throughputs. The congestion window size, W, is increased by one packet if $\delta < \alpha$ and is decreased by one packet if $\alpha > \beta$ (the default values of α and β are 1 and 3, respectively). In other words, having calculated δ, the sender updates W in accordance with

$$W = \begin{cases} W + 1, & \text{if } \delta < \alpha \\ W - 1, & \text{if } \delta > \beta \\ W, & \text{otherwise} \end{cases} \quad (2)$$

To detect and avoid congestion during the slow-start phase, Vegas doubles the window size every two RTT intervals rather than every RTT interval as it does in Reno. Furthermore, when receiving a duplicate ACK, the sender checks whether the difference between the current time and the sending time plus the *minRTT* of the relevant packet is greater than the timeout value. If it is, Vegas retransmits the packet immediately without waiting for the arrival of three duplicate ACKs. This modification resolves the problem that the sender never actually receives three duplicate ACKs and must therefore rely on the occurrence of the coarse-grain timeout and therefore yields a significant improvement in the throughput.

3 DESCRIPTION OF PROPOSED METHOD

The IEEE 802.11 link layer protocol supports reliable data to transfer locally while the TCP transport layer protocol supports reliable data to transfer end-to-end. This section proposes a collaborative approach for TCP Vegas enhancement.

3.1 Extension to IEEE 802.11 MAC

In standard IEEE 802.11 DCF schemes, whenever a node fails to transmit a frame (e.g. channel collisions), it retransmits that frame and then increases the value of the retry limit parameter by one. However, if the retry transmission exceeds a specified retry threshold, the IEEE 802.11 MAC protocol reports a link breakage and then discards the transmitting frame. To reduce the effects of link layer packet corrupt on the performance of TCP, this study extends the IEEE 802.11 DCF. Whenever a frame transmission failure occurs, a negative-ACK option is triggered by modified link layer protocol in accordance with the following function:

If the number of retry limit parameter exceeds a specified threshold {
If receiving of TCP ACK within the same flow {
Forward the negative-ACK option piggybacks using the reverse TCP ACK along the end-to-end path. }
}
Else {
Increase the contention window and then launch the backoff procedure.
}

In the extended protocol, if a TCP data frame is discarded after several retransmission attempts, the modified link layer protocol triggers a negative-ACK as the sequence number associates with the dropped packet and then piggybacks this option by a reverse TCP ACK in order to avoid increasing contention in the link layer. This cross-layer support from the link layer protocol ensures that the transport layer TCP protocol is aware of the transmission error in the link layer and can then react to this error in accordance with the wireless corruption information.

The extended link layer protocol provides additional information for natively wired TCP protocols to the wireless environment and for improving their performance. A major advantage of the proposed scheme is that it eliminates the requirement of forward nodes to cache unacknowledged packets for every TCP connection passing through it and is required to be fully supported at all of the wireless stations in the wireless network.

3.2 Modification to Vegas Congestion Control

To estimate the buffer currently available in each round-trip time interval, the expected bandwidth for TCP at the bottleneck link in the network is first derived. The queue length at the bottleneck is then approximately measured in each round-trip time interval from the inter-arrival times of the returned ACK messages. The available buffer size along the path between the sender and the receiver is then estimated from the derived value of the expected bandwidth and the measured value of the queue length. Finally, the TCP Vegas threshold parameters α and β are adjusted in accordance with the estimated buffer length, thereby, changing the size of the congestion window indirectly and consequently improving the network utilization. The details of the proposed Vegas transmission control scheme are described in the following.

Let *maxRTT* be the maximum measured RTT, and λ denote the bottleneck link bandwidth as calculated by the packet-pair scheme (i.e. the number of the acknowledged packets divided by the average ACK inter-arrival time) (Keshav, 1991). Let Q_m represent the measured buffer size in last RTT. According to the information obtained from last RTT, Q_m can then be calculated as:

$$Q_m = \left(\rho \cdot \left(\frac{W}{minRTT} - \frac{W}{maxRTT} \right) + (1-\rho) \cdot \left(\lambda - \frac{W}{maxRTT} \right) \right) \times minRTT \quad (3)$$

where ρ is a smoothing factor, and Q_m is obtained by the exponentially weighted moving average (EWMA) method.

Let Q_t denote the average observed buffer size which can be held in the end-to-end path. To prevent the estimated queue length from wild fluctuation, Q_t is configured by using the EWMA as:

$$Q_t = \nu \cdot Q_{t-1} + (1-\nu) \cdot Q_m \quad (4)$$

where the parameter, ν indicates the ratio of TCP sender transmission time to its measured round-trip delay time. The value of ν varies with the bandwidth utilization of the sender in the previous round-trip.

According to Eq. (4), the threshold values α and β are then adjusted as follows:

$$\alpha = \begin{cases} Q_t, & \text{if coutinue staying in stable status} \\ \alpha/2, & \text{if } W \text{ continue staying in stable status beyond a preset duration} \\ \alpha/2, & \text{if packet loss} \\ 1 \text{ (default)}, & \text{if coarse grained timeout} \end{cases} \quad (5)$$

where $1 \le \alpha \le Q_t - 2$

$$\beta = \begin{cases} m \cdot \alpha \\ 3 \text{ (default)}, \text{ if coarse grained timeout} \end{cases} \quad (6)$$

where $3 \le \beta \le Q_t$, and the default value of m is 1.5.

When receive an ACK, the modified Vegas regulates its transmission rate in accordance with the following function:

If receiving a new ACK {
Adjust the threshold parameters α and β according to Eqs. (5) and (6) and then regulates the congestion window size based on the following function that mentioned in Eqs. (1) and (2):

$$W = \begin{cases} W + 1, & \text{if } \delta < \alpha \\ W - 1, & \text{if } \delta > \beta \\ W, & \text{otherwise} \end{cases}$$

}
Else {
If receiving a triple-duplicate ACK, and the

next expected packet is not a "corrupted packet" {
Retransmit the next expected packet and set $W_t = W/2$ and $W = W_t$. Then modified Vegas enter the regular TCP Fast recovery phase. }
}
If receiving the ACK which contains a negative-ACK option {
Recording sequence number of the "corrupted packet" obtained from negative-ACK, and then retransmitting the corrupted packet.
}

In the proposed approach, the negative-ACK number of the corrupted packet is tracked using a circular queue. By inspecting the historical negative-ACK information, the modified TCP can clearly identify corruption losses in the wireless links and therefore determine whether it is necessary to invoke the regular congestion control mechanisms. Hence, if retransmitted packets are lost repeatedly as a result of transmission errors, the original packets can be retransmitted immediately upon receipt of an ACK with a negative-ACK option without waiting for timeout expiry or for the "round-trip" time to elapse as in conventional TCP error recovery schemes.

4 NUMERICAL RESULTS

In the simulations, the *ns*-2 (Network Simulator, NS-2) 802.11 MAC layer and transport layer modules were both extended to model the proposed cross-layer functionality. The parameter values used in the simulations are summarized in Table 1. The default sizes of RTS, CTS and ACK packets are assigned to 20 bytes, 14 bytes and 14 bytes, respectively. Furthermore, the two-ray ground reflection model (Rappaport, 1960) was used to simulate signal propagation in the wireless network.

Table 1: Default parameters for MAC and physical layers.

Parameter	802.11g
SLOT	9 μsec
SIFS	10 μsec
DIFS	28 μsec
PHY_{hdr}	192 bits
CW_{min}	32
CW_{max}	1024

In the following simulations, packet errors are introduced into the wireless link (indicated by dotted line) using the two-state Markov error model (Gilbert, 1960) (see Figure 2). The packet corruption rates P_G and P_B are set as 0.0005 and 0.001, respectively, and P_{GG}, P_{GB}, P_{BB} and P_{BG} are set to 0.9, 0.1, 0.85 and 0.15, respectively. The rate at which errors occur in the *GE* model depends on channel conditions.

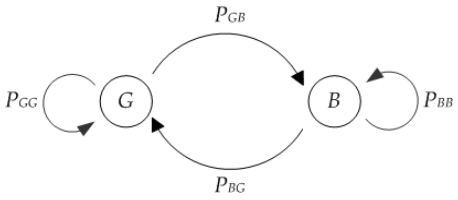


Figure 2: Two-state Markov error model.

Figure 3 gives the simulation topology when two TCP flavors (Reno and Vegas) coexist.

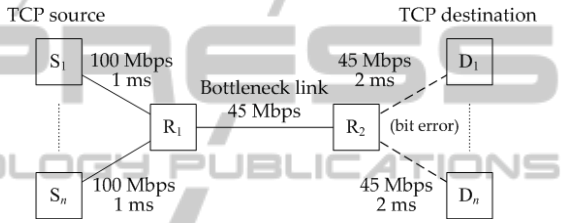


Figure 3: Simulation model with link-level errors.

Table 2 compares the proposed Vegas modification with the original Vegas. As shown in Table 2, packets transmitted on wireless channels are frequently affected by burst errors which cause the MAC layer to discard the frame, resulting in the loss of the corresponding packet in the TCP protocol. In this case, although Reno still has a better performance, it can be seen that the variation between mechanisms is decreased. Transmission rate of Reno is not able to increase stably because of bit error and provide Vegas more opportunities to obtain more bandwidth. From results shown in Table 2, it can be seen that the proposed Vegas modification decreases the variation between the two mechanisms and achieves better bandwidth utilization.

Table 2: Interaction of TCP in heterogeneous environments (with link-level bit error).

TCP flavors	**Average goodput**
Reno:Vegas	23.59 Mbps:16.53 Mbps
Reno:Modieied Vegas	19.04 Mbps:19.76 Mbps

To further examine the performance of the proposed scheme under a more realistic and complicated network environment, a series of simulations is conducted on randomly generated ad hoc network topologies. In these simulations, the

network region is 100 m × 100 m with 20 wireless stations, and TX_{range} and PCS_{range} are set to 40 m and 85 m, respectively. As illustrated in Figure 4, at the beginning of each simulation, wireless stations are randomly placed within this region, and a random ad hoc topology is constructed. The number of short-lived TCP connections varies from two to nine. Moreover, the TCP source and the TCP destination of each connection are randomly selected.

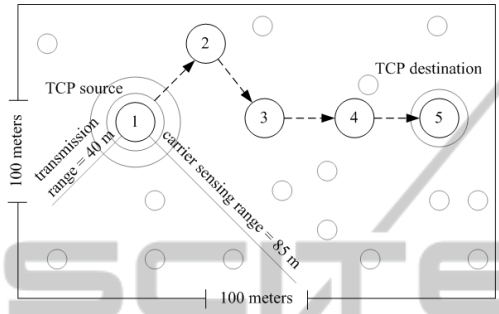


Figure 4: Illustrative random topology.

Table 3 presents the TCP goodputs of the proposed mechanism and other TCP flavors when the path lengths are 2-hop and 3-hop, respectively. As shown in Table 3, the TCP goodput is inversely proportional to the number of connections. This phenomenon is caused by the competition for the bandwidth among TCP connections and the carrier sense dependencies over wireless stations. Comparing the results in Table 3, it is obvious that the proposed Modified Vegas achieves a better performance than those of other TCP schemes.

Table 3: TCP average goodput comparison (Only the TCP goodput with established connection is calculated).

Number of connection	Reno	Vegas	Modified Vegas
2	11.58Mbps	11.51Mbps	14.52Mbps
3	9.76 Mbps	10.32Mbps	12.51Mbps
4	8.75 Mbps	8.74 Mbps	10.82Mbps
5	7.24 Mbps	8.20 Mbps	9.32 Mbps
6	6.53 Mbps	7.11 Mbps	8.90 Mbps
7	6.50 Mbps	7.04 Mbps	8.37 Mbps

5 CONCLUSIONS

This paper has proposed a cross-layer approach designated to enhance the performance of TCP Vegas in wireless networks. In the proposed approach, a cross-layer mechanism is employed in both the transport layer and the MAC layer to provide explicit corruption loss information. The results confirm that the proposed scheme has a number of key advantages compared to the conventional TCP, including a more efficient treatment on frequent transmission losses, a faster reaction to corruption losses, and the ability to distinguish between congestion errors and transmission errors and to take appropriate remedial action which is particularly advantageous for the deployment in heterogeneous wireless environments.

REFERENCES

IEEE Standard 802.11, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 1999.

IEEE Standard 802.16-2004, *Part 16: Air Interface for Fixed Broadband Wireless Access System*, 2004.

Paxson, V., Allman, M., Stevens, W., *TCP Congestion Control*, RFC 2581, 1999.

Floyd, S., Henderson, T., *The NewReno Modification to TCP's Fast Recovery Algorithm*, RFC 2582, April 1999.

M. Mathis, J. Mahdavi, S. Floyd, A. Romanow, TCP Selective Acknowledgement Options, RFC 2018, April 1996.

M. Jain, C. Dovrolis, End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput, IEEE/ACM Transactions on Networking, Vol. 11, No. 4, pp. 537 – 549, 2002.

R. Jain, A Delay-based Approach for Congestion Avoidance in Interconnected Heterogeneous Computer Networks, Computer Communication Review, Vol. 19, No. 5, pp. 56 – 71, 1989.

L. Brakmo, L. Peterson, TCP Vegas: End to End Congestion Avoidance on a Global Internet, IEEE Journal on Selected Areas in Communications, Vol. 13, No. 8, pp. 1465 – 1480, 1995.

R. S. Cheng, M. Y. Shih, C. C. Yang, Performance Evaluation of Threshold-based Control Mechanism for Vegas TCP in Heterogeneous Cloud Networks, International Journal of Internet Protocol Technology, Vol. 5, No. 4, pp. 202-209, March 2010.

J. Kim, B. Crow, I. Widjaja, P. Sakai, IEEE 802.11 Wireless Local Area Networks, IEEE Communications, Vol. 35, No. 9, pp. 116 – 126, 1997.

S. Keshav, A Control-Theoretic Approach to Flow Control, ACM SIGCOMM Computer Communication Review, Vol. 21, No. 4, pp. 3-15, 1991.

Network Simulator, NS-2, Available: http://www.isi.edu/nsnam/ns/.

T. S. Rappaport and T. Rappaport, Wireless Communications: Principles and Practice, 2nd Edition, Prentice Hall, 2001.E. N. Gilbert, Capacity of a Burst-noise Channel, Bell System Technical, Vol. 69, pp. 1253 – 1265, 1960.

E. N. Gilbert, Capacity of a Burst-noise Channel, Bell System Technical, Vol. 69, pp. 1253 – 1265, 1960.