

Research Article

Comparative Performance Evaluation of TCP Variants in WiMAX (and WLANs) Network Configurations

Konstantinos Tsiknas and George Stamatelos

Department of Electrical and Computer Engineering, Democritus University of Thrace, Vasilissis Sofias 12, 67100 Xanthi, Greece

Correspondence should be addressed to Konstantinos Tsiknas, ktsiknas@ee.duth.gr

Received 5 October 2011; Revised 24 December 2011; Accepted 2 January 2012

Academic Editor: Mohamed El-Tanany

Copyright © 2012 K. Tsiknas and G. Stamatelos. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

An important application for the IEEE 802.16 technology (also called WiMAX) is to provide high-speed access to the Internet where the transmission control protocol (TCP) is the core transport protocol. In this paper we study through extensive simulation scenarios the performance characteristics of five representative TCP schemes, namely, TCP New Reno, Vegas, Veno, Westwood, and BIC, in WiMAX (and WLANs) networks, under the conditions of correlated wireless errors, asymmetric end-to-end capabilities, and link congestion. The target is to evaluate how the above conditions would affect the TCP congestion control and suggest the best schemes to be employed in WiMAX networks.

1. Introduction

Broadband wireless access (BWA) is a candidate for the next-generation wireless communication technology. Worldwide interoperability for microwave access (WiMAX) is the organization behind interoperability and testing for the IEEE 802.16 Specifications [2]. For WiMAX to be able to realize the objective of high speed Internet access, it must effectively support the core transport protocol of the Internet that is transmission control protocol (TCP). Standard TCP congestion control is based on the reduction of its congestion window after a packet loss [3]. Although such behavior works fairly well in the wired networks, where packets losses are almost always caused by link congestion, it becomes rather inefficient when used for data transport in WiMAX networks. In the wireless environment, the possible reasons of packet loss include fading, temporary disconnections, and handovers. Even when some losses are compensated in Data Link Layer, a part of them still appears in Transport Layer for high Bit Error Rates (BER). TCP New Reno erroneously translates the causes of packet loss as congestion and consequently reduces its congestion window, thus exhibiting significant throughput degradation in these conditions.

Another phenomenon which significantly affects TCP performance is the network asymmetry. A network is

assumed to be asymmetric when its characteristics in one direction greatly affect its performance in the other [4]. Asymmetry disrupts the smooth flow of ACK in the reverse direction of traffic and subsequently the TCP ACK clocking mechanism of the TCP sender, causing timer expiration, and subsequent retransmissions, even though the packets may have correctly reached the receiver. In addition, frequent ACK delays result in timeout inflates, thereby causing reduced protocol responsiveness to packet losses. There are several forms of asymmetry, such as bandwidth asymmetry, access asymmetry, delay asymmetry, and packet loss asymmetry. All of them are inherent in WiMAX networks, particularly in best-effort class of service, where TCP typically operates, due to the contention for traffic among subscriber stations (SSs) and because the SS needs to receive a grant by the base station (BS) before sending any data [5]. Bandwidth asymmetry is also very often in WiMAX networks when time division duplex (TDD) is used. In TDD, the ratio of transmission between downlink (DL) and uplink (UL) direction can be adaptive. While high DL:UL ratios are good for large amounts of traffic in the downlink direction, they may cause bandwidth starvation in the uplink, which in turn affect the smooth flow of ACKs.

A lot of research has been made so far, aiming at suggesting ways to improve the efficiency of TCP in wireless

networks. However, there are not extensive comparative studies of TCP performance in WiMAX networks. In this work we evaluate some representative TCP congestion control schemes under various traffic scenarios, which include single and multiple TCP flows through WiMAX networks in the presence of wireless channel errors, network asymmetries, and various level of link congestion. The target is to find out the best performing TCP schemes and to suggest ways for further improvements. The paper is organized as follows: Section 2 reviews the following TCP variations: New Reno, Vegas, Westwood, and BIC; Section 3 describes the test environment for simulating the WiMAX networks; Section 4 presents the simulation results; Section 5 makes a comparison with the WLAN (WiFi) networks; the last section draws our conclusions.

2. Related Work

TCP New Reno [6] maintains two variables, the congestion window size (*cwnd*), which initially sets to 1 segment, and SS threshold (*ssthresh*). At the beginning of the TCP connection, the sender enters the *slow start* (SS) phase, in which it increases the *cwnd* by 1 segment for every ACK it receives. When *cwnd* reaches the *ssthresh*, the TCP sender enters the *congestion avoidance* (CA) phase, in which it increases the *cwnd* by $1/cwnd$ for every ACK it receives, in order to slowly probe the available network bandwidth. This linear growth ends when *cwnd* reaches the receiver's advertised window, or by the reception of 3 DUPACKs. In the latter case, TCP infers that packets were lost due to link congestion, and it reduces the *cwnd* by $1/2$ of its current value, in an attempt to prevent network collapse (*Fast Recovery*). This *additive increase multiplicative decrease* (AIMD) TCP's reactive congestion control and avoidance mechanism was proved incapable of handling efficiently mixed type packet losses happening in wireless heterogeneous networks [7].

TCP Vegas [8] abandons the linear window increase scheme of Reno and deploys a different congestion avoidance mechanism, attempting to estimate the level of network congestion before happening and tries to avoid it. Its decision is based on throughput measurements per *round trip time* (RTT), which reflect the network condition. Hence, Vegas calculates the difference between the *actual rate* (packets sent per RTT) and the *expected rate* (packet sent per best RTT) at the sender. If it is less than threshold *a*, this is an indication that the network resources are underutilized, and it increases its window by one segment. If the difference is greater than threshold *b*, this signifies that the network is about to experience congestion, and it decreases its window size by one segment to prevent it from happening. Otherwise, it keeps its congestion window the same, as it makes the best usage of network resources. Vegas advantage lies in its ability to make a more efficient use of the available bandwidth by avoiding congestion and the unnecessary retransmissions of dropped packets which caused by the linear growth of Reno. However, it still treats all packet losses as a sign of congestion, thus leading to unnecessary window decreases in wireless networks.

TCP Veno [9] combines Vegas and Reno in order to enhance TCP performance over Wireless networks. It keeps TCP Reno AIMD algorithm, and uses Vegas RTT measurements to estimate the level of congestion when an error occurs, in order to make the right decision about the origin of error and the extent of window decrease. If the measurements show that the network was about to experience congestion, Veno reduces its window by $1/2$ of its current value, like Reno, in order to relieve the network. If not, the protocol deduces that packet loss is not related to congestion, and it reduces its window less aggressively, to $4/5$ of its current value, in order to prevent unnecessary performance degradation. Veno was tested in [10] against handovers in simulated WiMAX networks, where it showed an improvement of roughly 5% over Reno.

TCP Westwood [11] makes no attempt to correct the problem of noncongestion packet loss in wireless networks solely like Veno, but rather to improve the efficiency of TCP in all heterogeneous networks. It estimates the network's bandwidth by properly low-pass filtering and averaging the rate of returning acknowledgment packets per RTT. It then uses this bandwidth estimate to adjust the *ssthresh* and the *cwnd* to a value close to it when a packet loss is experienced (*adaptive decrease*). In particular, when three DUPACKs are received, both the *cwnd* and *ssthresh* are set equal to the *estimated bandwidth* (BWE) times the minimum measured RTT (RTTmin); when a coarse timeout expires, the *ssthresh* is set as before, while the *cwnd* is set equal to one. The improvement of Westwood is a more realistic bandwidth estimation in comparison to TCP Vegas, which significantly increases TCP throughput over wireless links. TCP Westwood has also been tested in [10] against handovers in simulated WiMAX networks, where it shows an improvement of 16.5% over Reno.

Finally, *TCP BIC* (*Binary Increase Congestion control*) [12] aims at improving the performance of TCP for high speed networks with high latency. It abandons the Additive Increase concept of TCP Reno for a more aggressive window increase mechanism, which allows TCP to occupy faster the available network's bandwidth. This method, which is called *binary search increase*, defines a minimum window size W_{\min} as the speed at which no packet loss occurs, and a maximum window size W_{\max} , for which packet loss occurred. Since the best window value should be somewhere between those two, the algorithm repeatedly sets the current window size in the middle of both values and checks for feedback, in the form of packet losses. Based on the feedback, the midpoint is taken as the new W_{\max} if there is a packet loss, and as the new W_{\min} if not. The algorithm allows bandwidth probing to be more aggressive initially, when the difference between W_{\min} and W_{\max} is large. As this difference shrinks, then the rate of increase of the current window size become less aggressive, which allows for fewer packet losses (logarithmic increase). TCP BIC has been adapted as the default TCP protocol of Linux Operating System from version 2.6.8 to 2.6.18. TCP BIC has been evaluated in [12] against other high speed access techniques (STCP, HSTCP), where it was found to provide high link utilization and RTT fairness.

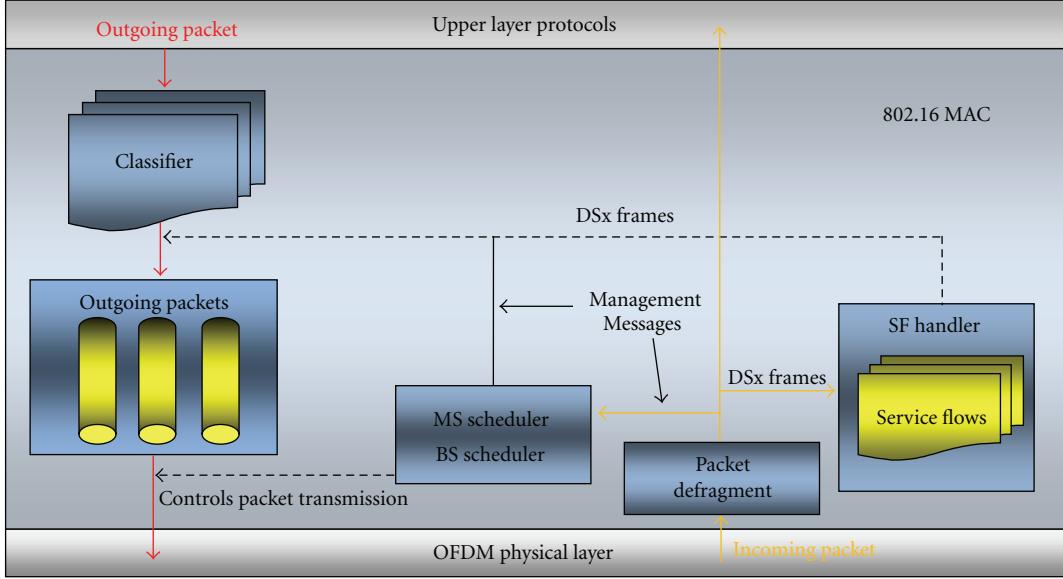


FIGURE 1: NIST 802.16 MAC Model of IEEE 802.16 Networks [1].

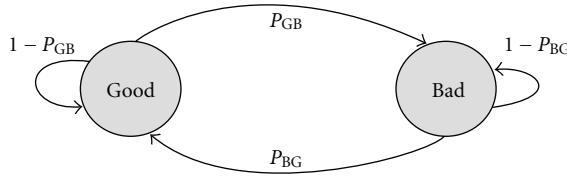


FIGURE 2: Two-state Markov Error model.

TABLE 1: IEEE 802.16 simulation parameters.

WiMAX and OFDM Parameters	
Quality of service	Best Effort
Channel bandwidth	7 MHz
Frame duration	5 msec
Modulation and coding	64 QAM 3/4
Cyclic prefix	1/16
Contention size	5
DL : UL ratio	1 : 1, 2 : 1, 3 : 1, 4 : 1, 5 : 1, and 6 : 1
Traffic source and other parameters	
TCP versions	New Reno, Vegas, Veno, Westwood, and BIC
TCP segment size	1024 bytes
Simulation duration	300 s

3. Simulation Environment

In this section we will present the test setup used for comparing the above TCP schemes. The traffic scenarios were implemented in network simulator 2 [13] by the use of the National Institute of Standards and Technology (NIST) models for WiMAX networks (see Figure 1).

Table 1 depicts the most important WiMAX and traffic parameters used in our simulation.

For modeling the wireless packet errors, the Gilbert-Elliott model (also known as two-state Markov model) was used. It is widely accepted for simulating the correlated errors in the wireless medium, which are caused by different phenomena, such as path loss, fading, noise, and interference [14]. According to this model, the process can be in either one of two states, namely the *good* state or *bad* state. The transition between the two states is modeled as a continuous-time Markov chain as depicted in Figure 2, where P_{GB} and P_{BG} are the transition rates from *good* to *bad* state and vice versa.

The mean sojourn time of the channel staying in a state can be approximated by $T_G = 1/P_{GB}$ and $T_B = 1/P_{BG}$, where T_G , and T_B are exponentially distributed. Very often the steady state probability of bad state (f_b) is expressed as the long-term fraction of the time the channel spends in that state [15]:

$$f_b = \frac{E[T_B]}{E[T_G] + E[T_B]}. \quad (1)$$

The metric used for evaluating the performance of TCP variants is the *Average Goodput*, which measures the amount of data bytes (without overheads) sent and successfully acknowledged, defined as:

Average Goodput

$$= \frac{\sum_{n=1}^N \frac{\text{MaxAckSeqNo}_n \times \text{MSS}_n}{\text{TotalTransmissionTime}}}{N}, \quad (2)$$

where MaxAckSeqNo represents the maximum ACK number of a connection n that is received by a sender, MSS is the maximum segment size of the connection n , and N is the total number of connections. Higher goodput means better TCP performance. *Fairness* is another important metric for

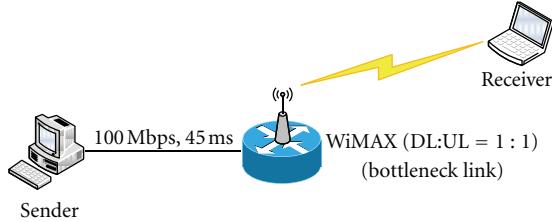


FIGURE 3: A single TCP flow in a WiMAX network.

evaluating that all TCP connections fairly utilize the network bandwidth. We use Jain's Fairness Index [16], defined as:

$$\text{Fairness Index} = \frac{\left(\sum_{i=1}^N S_n\right)^2}{N \cdot \sum_{i=1}^N (S_n)^2}, \quad (3)$$

where S_n is the Throughput of flow n . The range of fairness is between 1 and $1/N$, where 1 corresponds to the fairest value.

4. Simulation Results

In this section we present our simulation scenarios and discuss the results obtained for WiMAX networks. Several scenarios are considered to highlight the effects of wireless channel errors, asymmetry, and link congestion on TCP performance.

Scenario 1 (Effect of Wireless Channel Errors). This scenario includes data transfer through a single TCP connection over a wired link of bandwidth 100 Mbps and delay 45 msec and a WiMAX network (see Figure 3).

The purpose of this scenario is to find out whether the evaluated TCP congestion control algorithms can deal with the wireless channel errors and adaptively adjust their congestion window, in order to achieve the best possible performance.

Measures of goodput are received for all compared approaches for various packet loss rates, ranging from 0.001% to 5%, and the results are displayed in Figure 4. A first observation is that the goodput of all TCP variations decreases as the wireless packet loss rate increases, with Westwood offering the best results (96% higher goodput than Reno for PER = 0.001). Westwood's advantage lies on its bandwidth estimator (BWE) mechanism, which, by properly averaging and filtering the receiving samples per RTT, it provides a good judgment of the network's bandwidth and minimizes the effect of wireless channel errors (see Figure 5). Regarding BIC, our measurements show that it also offers a small performance improvement over Reno (about 18% for the same PER). This is due to its binary increase algorithm, which by prompting for fast window inflates it results in a better channel utilization (see Figure 5).

On the opposite, Vegas and Veno present lower performance than Reno for higher loss rates. Since they both attribute to congestion the increased RTT caused by the packet errors, they constantly keep their windows at very low values than the available bandwidth (see Figure 5).

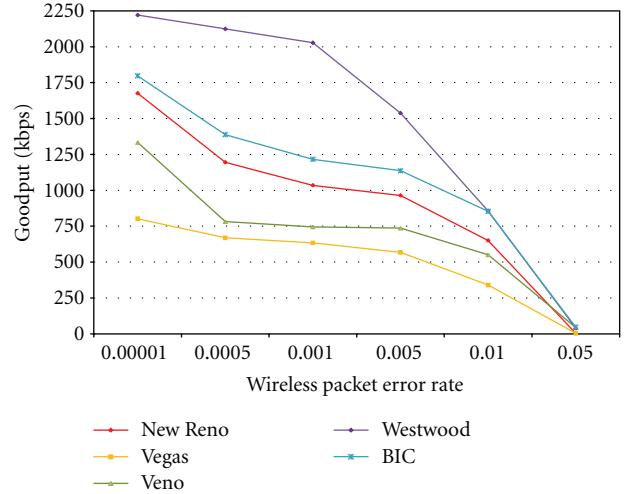


FIGURE 4: Goodput versus PER in WiMAX for the Scenario 1.

Scenario 2 (Effect of Wireless Channel Errors and Link Congestion). The network setup of Figure 6 resembles a possible structure of a WiMAX system integrated with a wired network.

In addition, for simulating the bidirectional cross Internet traffic of a real network, on/off Pareto flows [17, 18] are injected in the wired portion, from C1 to C2 and C4 to C3. The rate of the flows is constant and equal to 20 Mbps and 0 Mbps during the on and the off intervals, respectively, while the mean duration in on and off states is 500 msec. The target of this scenario is to evaluate the TCP congestion control algorithms under a combination of correlated errors and congestion of the wired network. Figure 7 displays the results.

As expected, the goodput of all evaluated TCP variations degrades in comparison with Scenario 1, due to the combination of correlated errors and link congestion. However, our measurements show that this combination affects the evaluated protocols quite differently. BIC presents a small performance gain over Reno (approximately 10% for PER = 0.001), whereas the gain for Westwood is high (108% for PER = 0.001). This indicates that the BWE of Westwood works efficiently for low PER. As PER increases, however, this advantage is reduced rapidly and becomes negligible for PER greater than 0.005. In these conditions, the efficiency of its BWE is disrupted, causing inaccurate bandwidth estimations and low performance. Regarding TCP Vegas and Veno, they both present low goodput values; the combination of congestion and wireless channel errors provokes further increases in their RTT, which in turn reduces their transmission rates.

In order to evaluate TCP variants' reaction to congestion, we simulate them for different buffer sizes (involving the bottleneck link between R_0 and R_1). It is well established that then smaller the buffer size of the intermediate routers, then often packets are dropped, causing congestion. On the other hand, if the buffer size is too long, the packets may stay longer in the intermediate routers, forcing congestion on TCP due

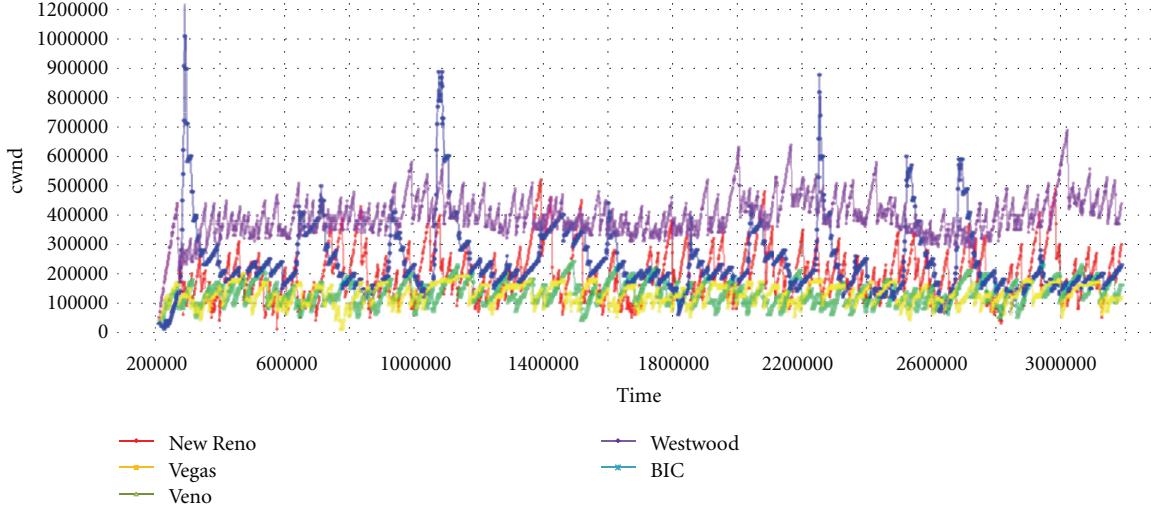


FIGURE 5: Variation of congestion window versus time (PER = 0.001).

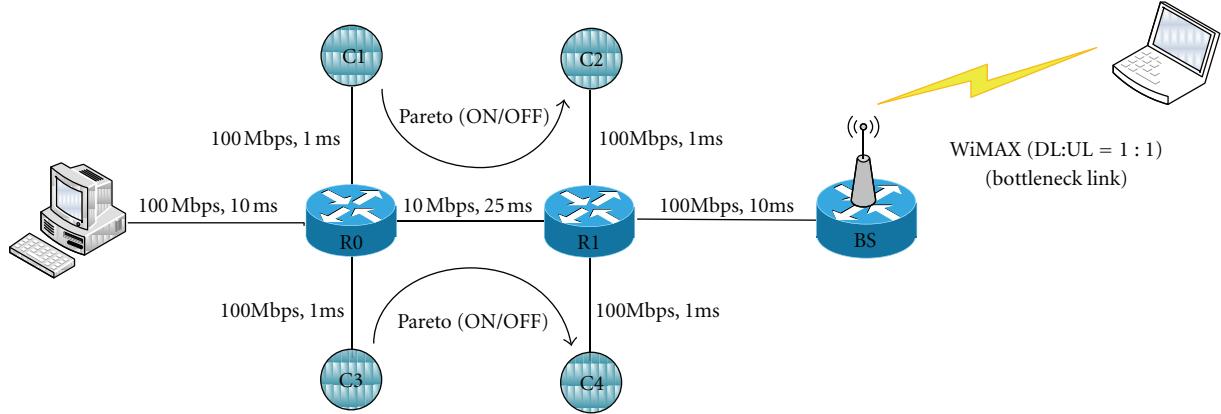


FIGURE 6: A single TCP flow in a heterogeneous WiMAX network.

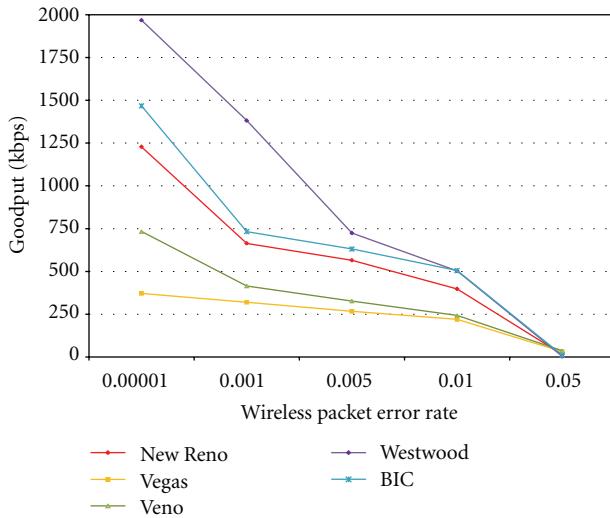
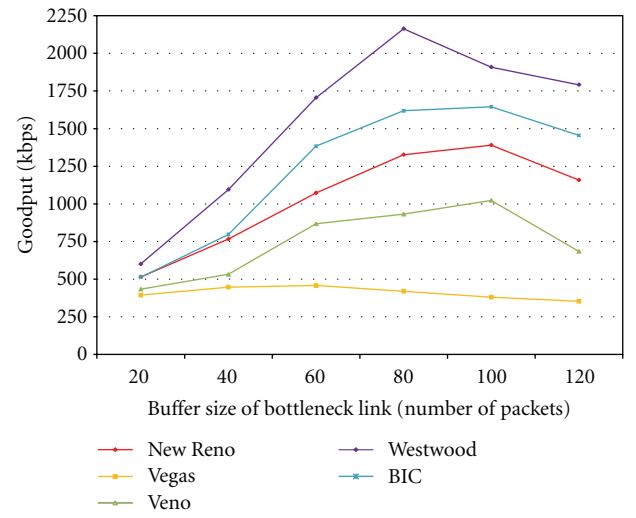


FIGURE 7: Goodput versus PER in WiMAX for the Scenario 2.

FIGURE 8: Goodput versus queue size of the bottleneck link in WiMAX for Scenario 2 (PER = 10^{-5}).

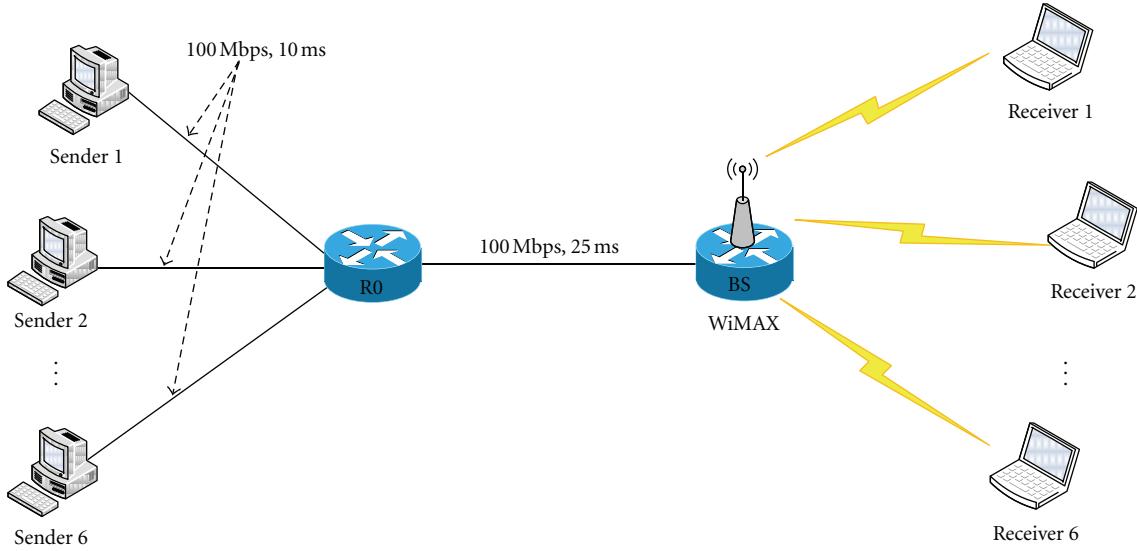


FIGURE 9: Multiple TCP flows in a WiMAX network.

to its *round trip timeout (RTT)* timer expiration. The “ideal” buffer size for TCP must be less than the *bandwidth delay product (BDP)* of the link [19].

The goodput values for various congestion levels are plotted in Figure 8. According to the measurements received, all TCP variants are affected by the congestion, but not to the same extent. TCP Westwood gives the best results; by adjusting its window to a value close to the BDP of the link, instead of “blindly” halving it like Reno, it maintains high transmission rates after a congestion episode, resulting in better link utilisation and performance. The highest goodput occurs when the buffer size of the bottleneck link is about 80 packets. From that point and beyond Westwood’s performance gradually degrades, because of the long queuing delays introduced by the Pareto and TCP flows.

Also BIC responds quicker than Reno to congestion thanks to its logarithmic increase mechanism, which fast seizes the available bandwidth. On the other hand, our results show that Veno and Vegas show poor performance levels for the whole range of congested link tested (see Figure 8).

Scenario 3 (Effect of Asymmetry). For evaluating the effects of asymmetry on TCP performance, the following scenario is examined for all compared approaches. In Figure 9, a total of 6 TCP flows operate in a WiMAX network offering best-effort service to receivers 1–6.

In this configuration the bottleneck is the wireless link. Apart from the correlated errors, it also presents *access* and *bandwidth asymmetry*, due to the different transmission ratio between the DL and UL path (see Section 1). The results for DL:UL ratio equal to 3:1 are shown in Figure 10.

As measurements show, Westwood and BIC are the best performing TCP variations, presenting up to 58% and 54% better goodput correspondingly over Reno, per each TCP connection and for PER = 10^{-4} . On the other hand, Veno and

Vegas exhibit inferior performance, due to the long RTTs the asymmetry provokes.

Regarding Fairness, our measurements show that the evaluated TCP protocols have fairness index close to 1 for error rates up to 1%, thus ensuring a fair distribution of the available resources among the TCP connections sharing the common channel, while the fairness index becomes low when the PER is greater than 5%. Figure 11 depicts the results for DL:UL ratio equal to 3 to 1.

In order to examine in detail how the different levels of asymmetry affect TCP, we repeated the same scenario for the best performing TCP protocols, that is, Westwood, BIC and Reno and for various DL:UL ratios, ranging from 1:1 to 6:1, corresponding to values 0.50 to 0.857 in Figure 12.

For DL:UL ratios equal to 2:1, the average goodput of the three TCP variants is increased by approximately 20% in comparison to the 1:1 case. For higher DL:UL ratios, however, their goodput still increases, but with lower rate (approximately 12% increase rate in the range 2:1 to 5:1). As the DL:UL ratio increases, there is more available bandwidth in the downlink direction, and TCP continuously inflates its window to make use of it. As congestion window increases, the traffic load of ACKs in the reverse channel grows larger, resulting in high competition for uplink bandwidth. For high DL:UL ratios, where the number of uplink slots is reduced, ACKs are delayed or lost, causing packet retransmissions.

Another important finding is that the performance of BIC is not affected by the asymmetry to that extend as the other TCP variants, and presents the best goodput over the others for DL:UL ratio equal to 6:1 (see Figure 12). This is due to its binary increase mechanism, which allows for a more aggressive bandwidth seizure of the downlink channel after a retransmission. With regard to Fairness, the measurements show some noticeable fluctuations for

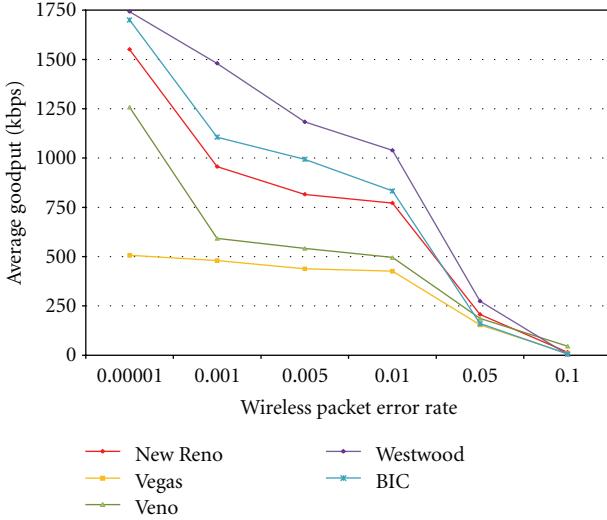


FIGURE 10: Goodput versus PER for the Scenario 3 (DL: UL = 3 : 1).

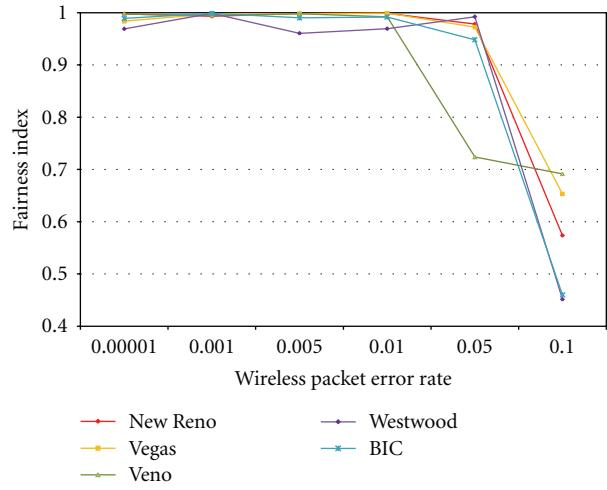


FIGURE 11: Fairness versus Packet error rate (DL: UL = 3 : 1).

Westwood, for high DL:UL ratios, due to the bandwidth asymmetry (see Figure 13).

5. Comparison with Wlans

For comparison with the Wireless LAN, the CMU Monarch implementation of IEEE 802.11b (also called WiFi) was used in ns-2 simulator [20]. The most important WiFi parameters used in our simulations are listed in Table 2.

We are interested in the effects of wireless errors and link congestion. Therefore, we have implemented Scenario 1 (see Figure 3) and Scenario 2 (see Figure 6) in WiFi networks. The third scenario, for testing the effects of bandwidth asymmetry (see Figure 9), is not applicable in the most commercial WiFi applications (i.e., IEEE 802.11b), since it requires support of TDD. Our simulation results for Scenarios 1 and 2 are demonstrated in Figure 14 and Figure 15 correspondingly.

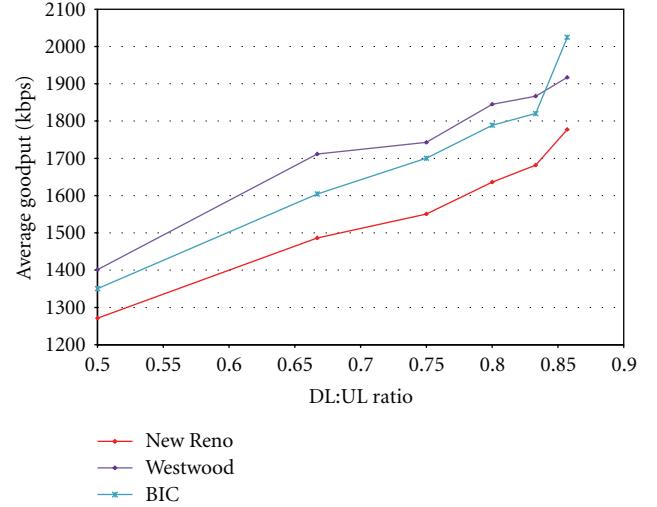
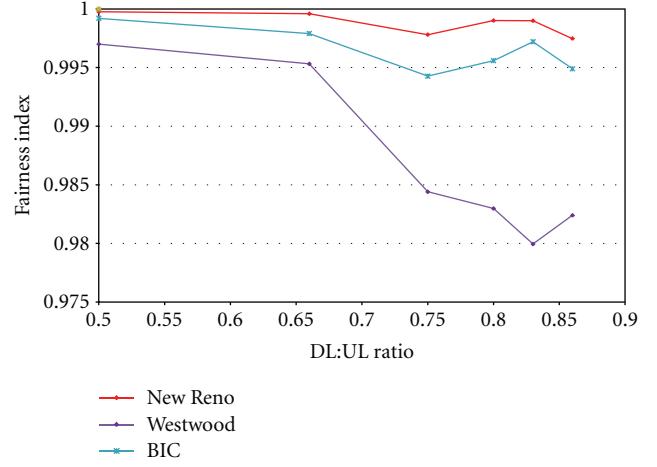
FIGURE 12: Average Goodput for various DL:UL ratios (PER = 10^{-5}).FIGURE 13: Fairness for various DL: UL ratios (PER = 10^{-5}).

TABLE 2: IEEE 802.11b simulation parameters.

IEEE 802.11b (WiFi) parameters	
Data rate	11 Mbps
RTS threshold	3000 bits
Preamble length	72 bits
Slot time	20 μ sec
SIFS	10 μ sec
DIFS	50 μ sec

As we observe from Figure 14, the evaluated TCP protocols achieve similar goodput levels for low packet error rates. This behavior is different in comparison with scenario 1 in WiMAX (see Figure 4). As PER increases, however, they start varying. For instance, TCP Westwood and BIC present on average about 35% and 9% higher goodput than Reno correspondingly, for PERs ranging between 0.01 and 0.1. With regard to Scenario 2, Figure 15 shows that the

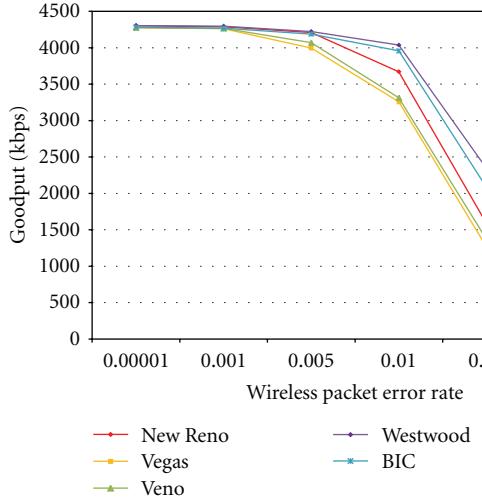


FIGURE 14: Goodput versus PER for Scenario 1. A single TCP flow in a WiFi network.

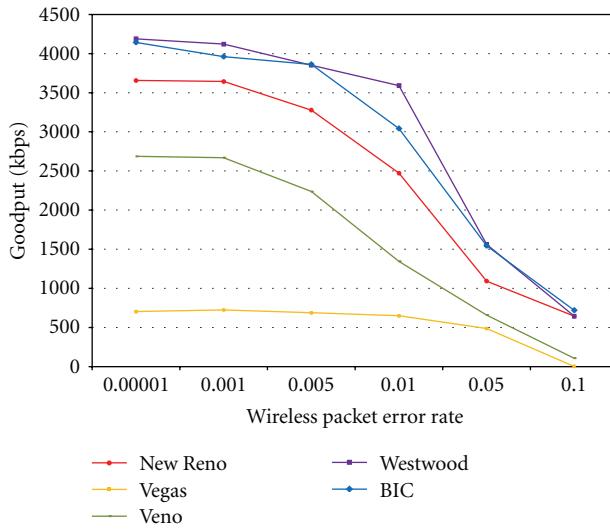


FIGURE 15: Goodput versus PER for Scenario 2. A single TCP flow in a heterogeneous WiFi network.

congested link in wired part of the network degrades Reno, Vegas and Veno performance, whereas Westwood and BIC are not affected very much for low PERs (less than 0.01). For higher PER however, their goodput drops abruptly. This behavior is similar with the corresponding WiMAX scenario (see Figure 7).

Finally, Figure 16 displays how the TCP variants react to congestion in the fixed part of the WiFi network for various congestion levels. As our results demonstrate, Westwood offers high goodput levels for the whole range of the congested link tested. Its unique property to adaptively adjust its transmission rate depending on the actual network conditions makes Westwood a very efficient transport protocol for very different congested environments. Also BIC offers high goodput levels, while Vegas and Veno do

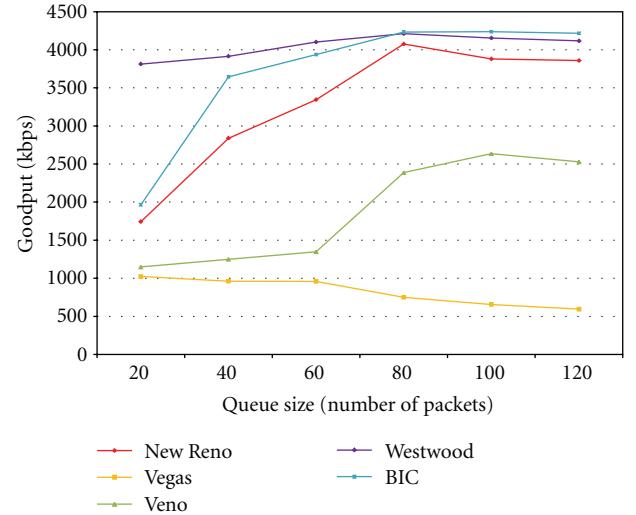


FIGURE 16: Goodput versus queue size of the bottleneck link in WiFi for the second scenario (PER = 10^{-5}).

not perform very well (similar results with WiMAX, see Figure 8).

6. Conclusion

In this paper, we have evaluated through extensive simulations the performance characteristics of various TCP schemes in WiMAX networks, by taking into account the effect of wireless channel errors, link congestion in both forward and reverse directions, and asymmetry. We saw that wireless errors, link congestion (and their combination) affect the TCP variants quite differently. We also observed that bandwidth asymmetry improves TCP performance up to a certain threshold value; in high asymmetric conditions, it results in low goodput values, due to the ACK delays (or losses) and the subsequent packet retransmissions it causes.

Overall, our measurements show that Westwood and BIC offer the best performance over the rest TCP variants in all cases studied. In particular, Westwood outperforms in WiMAX networks with wireless channel errors (up to 96% better goodput over Reno for PER = 0.1%) and in a heterogeneous WiMAX networks with congested wired links and low packet error rates (up to 108% goodput gain over Reno for PER = 0.01%). For BIC, there is a gain between 7% and 18% over Reno in all cases considered, while it presents the best performance in high asymmetric conditions. On the opposite, Veno and Vegas offer reduced performance levels in all examined configurations and therefore they are not suggested as transport protocols in WiMAX networks. Similar results were received in WiFi networks, where Westwood and BIC offer the best performance in all scenarios considered involving wireless errors and link congestion.

Regarding Fairness, both TCP Westwood and BIC offer a fair sharing of the available resources among TCP flows. Since the congestion control mechanisms of TCP Westwood and BIC are different, it is quite likely that a hybrid solution

combining the *binary increase (BI)* in congestion avoidance phase, with the *adaptive decrease (AD)* in fast recovery phase, could be an ideal candidate in WiMAX networks. Such a *binary increase adaptive decrease (BIAD)* paradigm will benefit by both the quick window expansions of BIC and by the appropriate window adaptations of Westwood, thus offering in overall a better performance in WiMAX networks.

Acknowledgment

This Work was supported by a THALES-CYBERSENSORS Research grant. The authors are with the Democritus University of Thrace.

References

- [1] R. Rouil, “NIST WiMAX network simulator,” Tech. Rep., NIST, 2007.
- [2] *IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, 2004.
- [3] V. Jacobson, “Congestion avoidance and control,” in *Proceedings of the (SIGCOMM '88)*, pp. 314–329, Standford, Calif, USA, August 1988.
- [4] H. Balakrishnan, V. Padmanabhan, and R. H. Katz, “The effects of asymmetry on TCP performance,” in *Proceedings of the 3rd ACM/IEEE Mobicom Conference*, Budapest, Hungary, September 1997.
- [5] A. Eshete, A. Arcia, D. Ros, and Y. Jiang, “Impact of WiMAX network asymmetry on TCP,” in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '09)*, pp. 1706–1711, Budapest, Hungary, April 2009.
- [6] S. Floyd, T. Henderson, and A. Gurtov, *The NewReno Modification to TCP's Fast Recovery Algorithm*, Request for Comments: 3782, Network Working Group, 2004.
- [7] F. Lefevre and G. Vivier, “Understanding TCP behaviour over wireless links,” in *Proceedings of the Symposium on Communications and Vehicular Technology*, pp. 123–130, 2000.
- [8] L. S. Brakmo and L. 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.
- [9] C. P. Fu and S. C. Liew, “TCP Veno: TCP enhancement for transmission over wireless access networks,” *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 2, pp. 216–228, 2003.
- [10] T.-W. Kim, S.-H. Chung, I.-S. Yoon, and J.-S. Kim, “Effects of handover on TCP congestion control algorithms over mobile WiMAX,” in *Proceedings of the 5th IEEE Consumer Communications and Networking Conference (CCNC '08)*, Las Vegas, Nev, USA, January 2008.
- [11] S. Mascolo and C. Casetti, “TCP Westwood: bandwidth estimation for enhanced transport over wireless links,” *Journal ACM Mobicom*, 2001.
- [12] L. Xu, K. Harfoush, and I. Rhee, “Binary increase congestion control (BIC) for fast long-distance networks,” in *Proceedings of the IEEE Conference on Computer Communications—23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, pp. 2514–2524, March 2004.
- [13] Network Simulator Ns-2, 2010, <http://www.isi.edu/nsnam/ns>.
- [14] H. S. Wang, “On verifying the first-order Markovian assumption for a Rayleigh fading channel model,” in *Proceedings of the 3rd Annual International Conference on Universal Personal Communications (ICUPC '94)*, pp. 160–164, San Diego, Calif, USA, October 1994.
- [15] K. Xu, Y. Tian, and N. Ansari, “Improving TCP performance in integrated wireless communications networks,” *Computer Networks*, vol. 47, no. 2, pp. 219–237, 2005.
- [16] R. Jain, W. Hawe, and D. Chiu, “A quantitative measure of fairness and discrimination for resource allocation in shared computer systems,” Tech. Rep. DEC-TR-301, 1984.
- [17] V. Paxson and S. Floyd, “Wide area traffic: the failure of Poisson modeling,” *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 226–244, 1995.
- [18] M. E. Crovella and A. Bestavros, “Self-similarity in world wide web traffic: evidence and possible causes,” *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 835–846, 1997.
- [19] C. Villamizar and C. Song, “High performance TCP in ANSNet,” *SIGCOMM Computer Communications Review*, vol. 24, no. 5, p. 4560, 1994.
- [20] “Cmu monarch extensions to ns,” September 2009, <http://www.monarch.cs.cmu.edu>.

