Improving Performance of TCP with Efficient MIMO-based MAC

Ankit Jindal and Anirudha Sahoo
Department of Computer Science and Engineering
Indian Institute of Technology, Bombay, Powai, Mumbai, India 400076.
email: {ankitjindal, sahoo}@cse.iitb.ac.in

Abstract

Wireless adhoc network is an attractive solution to provide connectivity between multiple devices without any support of infrastructure. But wireless networks typically have low throughput. Recent advances in MIMO technology promises to provide higher throughput in wireless network. But, the underlying MAC protocol should be suitably modified to exploit the MIMO functionality. In this paper, we apply basic MIMO concepts to 802.11 based multihop wireless network at the MAC layer. In order to take advantage of this MIMO based MAC, we propose corresponding modification at the TCP layer in terms of congestion window limit (CWL). Our simulation results show that running TCP with the modified MIMO-based MAC improves TCP performance. Further, from simulation data we determine that setting CWL to $1/2.8$ *of round trip hop count results in better TCP performance.*

1. Introduction

Wireless adhoc networks are attractive solution for providing connectivity amongst users in the absence of networking infrastructure. Nodes in this kind of network cannot have centralized control and hence use distributed medium access protocols such as IEEE 802.11 to communicate with each other. One of the main problems with wireless adhoc networks has been the poor performance of TCP applications as compared to wired networks. Since 802.11 is a contention based protocol and due to hidden node problem, the number of nodes along a TCP path that can simultaneously transmit is less than that in wireline network. This leads to lower throughput of TCP in 802.11 based adhoc networks. Past studies [1, 3–5, 7, 11] have shown that TCP performs poorly in wireless adhoc network, since its performance is highly dependent on the performance of the layers below it. Various issues like high contention and mobility degrade the performance of TCP.

Most of the work in this field are designed for single antenna transmitter and single antenna receiver system. Recent advances in the multiple-input multiple-output (MIMO) technology in antenna system promises to provide higher throughput without increasing transmit power. This new technology requires suitable changes in the MAC layer to improve the network performance. There have been several research work reported at the MAC layer for MIMO-based wireless network [2, 6, 9]. But the higher layer protocols like TCP has to make appropriate changes to percolate the improved performance at the MAC layer up to the TCP layer. In fact, without these changes, the performance of TCP may deteriorate compared to regular (non-MIMO) wireless network.

In this paper, we focus on improving the TCP performance over MIMO links. First, we study basic concepts in some of the MIMO-based MAC protocols and apply these concepts to MIMO-based multihop wireless network. Then, we look into the ways to improve TCP performance over this MIMO-based MAC protocol. It is well known that, in wireless adhoc networks, increasing the TCP window size beyond certain limit degrades its performance. The recent research reported in [1] shows that, in normal wireless network with IEEE 802.11 based MAC, setting the TCP congestion window limit (CWL) to $1/5$ of round trip hop count improves its performance significantly. However, the limit calculated with IEEE 802.11 based MAC might degrade the TCP performance over MIMO links. Since MIMO with efficient MAC protocol, offers more parallel communications in the same collision domain, modification of this limit is required in order to have improved TCP throughput. This paper present the relevant changes required at the TCP layer in terms of congestion window limit, with the efficient MIMO based MAC layer. Our simulation results show that running TCP with the proposed congestion window limit over the modified MIMO based MAC improves its performance. Further, from simulation data we determined that setting CWL to $1/2.8$ of round trip hop count results in better TCP performance.

The rest of the paper is organized as follows. Section 3 first presents basic concepts of MIMO-based MAC protocols and then the application of these concepts in multihop wireless networks. In Section 4, we propose modifications required at the TCP layer with the efficient MIMO based MAC. In Section 5, we present our simulation setup and results which justify our proposed modification at the TCP layer. Finally, we conclude this paper in Section 6.

2. Related Work

There has been a lot of research reported in literature to improve performance of TCP in wireless adhoc networks [1, 3–5, 7]. The basic conclusion out of most of these studies is to allow TCP window to hover around the value that allows maximum spatial reuse of the channel. In [3], the authors proposed two techniques, RED and adaptive pacing, such that average TCP window settle down at optimal point. In another study [12], the authors reported instability of window based congestion control algorithm and proposed a rate based end to end congestion control scheme. A leaky bucket rate controller is used in the TCP layer to control the TCP sending rate. The feedback from the bottleneck nodes along the path is used to control the sending rate.

In another study [1], the congestion window limit is set to $1/5$ of round trip hop count in order to improve TCP performance. However, the limit is calculated for regular IEEE 802.11 based MAC. But when nodes are MIMO-enabled, IEEE 802.11 based MAC results in under utilization of the network, since it does not make use of multiple antennas at transmitter and receiver. In order to fully utilize MIMO capabilities, there have been various MAC protocols proposed [2,6,9,10] in the literature. These MAC protocols are designed to have more parallel communications in the same collision domain. The key idea behind these proposed MAC protocols is the selection of the antenna weights and exchanging this information with neighboring nodes. In [6], the authors proposed the use of two channels, i.e., control channel and data channel. The control channel is used for exchanging control information like RTS, CTS, weights etc, while data channel is meant for actual data transmission. [9] have a different design of MAC protocol, but achieves the same goal as that of [6] without having a separate control channel.

3. Efficient MIMO based MAC Protocol for multi-hop communication

There have been different types of MAC protocols proposed for MIMO systems. Some of them take advantage of physical layer phenomena like diversity gain and spatial multiplexing gain. There are other MAC protocols which assign appropriate weights to the antennas to have multiple simultaneous communication in one collision domain. In this paper, we consider MAC protocol of latter type.

To understand why there is a need for a more efficient MAC protocol for MIMO based system, let us look at Figure 1 and assume that the transmission and interference ranges of the nodes are the same. The dashed circles show the transmission ranges of the nodes. When node A is transmitting to B, node C cannot transmit to D, since that will lead to collision at B (this is classical *hidden node* problem). This means in every three hops (or nodes) there can be only one communication in the forward direction. Hence, a paths with h nodes can have at most $h/3$ number of simultaneous transmissions. The solid circles in Figure 1 indicate a collision domain or a transmission domain. But if the wireless nodes are MIMO-based, then C can transmit to D when A is transmitting to B. This is possible because, using MIMO technology unintended signal reaching B from C can be nullified (details explained in latter section). This is shown in Figure 2. Hence, in MIMO based system, there can be one communication in every two hops, i.e., there can be $h/2$ number of simultaneous communications possible in a path of h hops.

It is easy to perceive that when interference range is more than transmission range (but less than twice the transmission range), regular 802.11 MAC allows $h/4$ simultaneous transmission. However, in case of MIMO-based system the number of simultaneous transmissions remains same as the previous case (when transmission and interference range is the same), because MIMO systems have the capability to nullify the interference from neighboring nodes.

Thus, if traditional 802.11 MAC is used over MIMO-based PHY, then it cannot exploit the capabilities provided by MIMO. Hence, an efficient MIMO based MAC protocol is required which will potentially increase the throughput because of increase in number of simultaneous transmissions. There are many MIMO based MAC protocols proposed in the literature [2, 6, 9]. These protocols are designed to take advantage of multiple antennas present at the transmitter and the receiver. Next, we start with a description of basic concept behind these MIMO based MAC protocols which provides more efficient multihop communication.

3.1. Basic Concept

We represent the MIMO channel by a $N \times N$ matrix H, where N is the number of antennas at both the transmitter and the receiver. Each entry h_{ij} of the matrix H represents the complex gain between the i^{th} transmitter antenna and the j^{th} receiver antenna. When a signal $s(t)$ is to be transmitted from the sender, the beamformer at the transmitter sends the weighted version of the modulated signal $s(t)$ through each antenna. Similarly, the receive beamformer applies weights to the received signal on each receive antenna, which is then demodulated to form the resultant receive signal. Let the weights of the transmitter antennas w_T be represented by a $N \times 1$ vector and the corresponding weights at the receiver be a $N \times 1$ vector denoted by w_R. Then the signal received at the i^{th} antenna is given by

$$x_i(t) = s(t) \sum_{j=1}^{N} w_{T_j} h_{ji}$$

The receive beamformer combines the $x_i(t)$ to obtain the beamformer output $r(t)$

$$r(t) = \sum_{i=1}^{N} w_{R_i} x_i(t)$$

Thus, $r(t) = s(t) w_T^T H w_R$, where w_T^T is the transpose of transmitter weight vector w_T. So, the complex gain experienced by $s(t)$ as it passes through transmit beamformer, the wireless channel and the receive beamformer is equal to $w_T^T H w_R$. Hence, transmit and receive weight vectors can be chosen appropriately to receive the transmit signal with a certain gain or to completely nullify the transmit signal. Obviously, the weight vectors play an important role in order to accept the signal of intended transmitter and reject the signal of interfering transmitters. The weight vectors should be set based on whether the receiver wants to receive the signal or wants to nullify the signal, as explained below.

- If w_R of the intended recipient of the message is fixed, then w_T should be chosen such that $w_T^T (H w_R) = 1$.
- If a receiver (say r_A) having weight vector w_R is already in communication with a transmitter (say t_A), and another transmitter (say t_B) in the communication range of receiver r_A wants to communicate with receiver r_B then the weight vector w_T of transmitter t_B should satisfy $w_T^T (H w_R) = 0$, so that it does interfere with the ongoing communication of r_A.
- If a transmitter (say t_A) having weight vector w_I is already communicating with a receiver (say r_A) and a receiver (say r_B) which is in the communication range of t_A wishes to receive from some other transmitter (say t_B), then r_B should choose its weight vector w_R such that $(w_I^T H) w_R = 0$ and $(w_T^T H) w_R = 1$, where w_T is the weight vector of transmitter t_B.

Hence, it is clear that in a MIMO-based 802.11 system, unintended signal received by the receiver can be nullified by appropriately choosing the weights of transmitter and receiver. This feature provided by MIMO can be exploited at the MAC layer in order to provide multiple communications in the same collision domain.

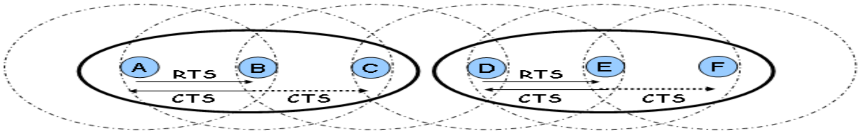

Figure 1: Illustration of Collision Domains and Number of Simultaneous Transmissions in regular IEEE 802.11 MAC

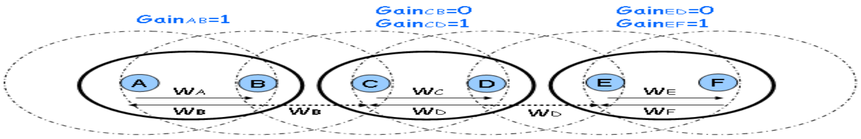

Figure 2: Illustration of Collision Domains and Number of Simultaneous Transmissions in MIMO-based IEEE 802.11 MAC

3.2. Modified MAC for MIMO based IEEE 802.11 Systems

Since multiple set of communication can happen in a MIMO-based system in a single collision domain, regular 802.11 MAC should be modified to take advantage of the MIMO PHY so that throughput of the network can be increased. Towards this goal we apply the basic concepts of MIMO based MAC protocol to multihop 802.11 based wireless network.

Before we formally present the modified MAC, we explain the main idea behind the protocol through the following example. Consider the chain topology of Figure 2. Whenever, node A wants to transmit data to node B, it transmits *RTS* using the *default weight vector*. Let us denote this by w_A. Default weight vector is the weight vector that is used by a transmitter, when it is the first transmitter to communicate in a given collision domain. The receiver upon receiving the message, determines the weight vector of the transmitter and chooses a weight vector, w_B, which it would use for reception. Normally, B can set $w_B = h^T$, where $h = w_A^T H$. Then B responds with a *CTS* message using its weight vector. The same weight vector w_B is used by B to receive data packet and to send an *ACK*. After the *RTS/CTS* handshaking, node A sends and node B receives data frame using the weight vector w_A and w_B respectively. Figure 2 illustrates this basic operation.

Now, when A and B are communicating, if C wants to start another communication with D, then it has to choose its weight vector carefully. C would have received the *CTS* sent by node B and hence knows the weight vector w_B of node B. C must select its weight vector, such that it does not cause interference at node B. So C should select its weight vector w_C such that the complex gain at B due to transmission of C is zero, i,e., $w_C^T(Hw_B) = 0$. C send the *RTS* to node D using this weight vector w_C. Thus, using appropriate weight vector multiple communication can happen in a single collision domain.

Now we formally present the weight selection of transmitter and receiver in a MIMO-based 802.11 network. Every node in the network keeps a record of weight vectors whenever it overhears an RTS or an CTS.

3.2.1. Transmitter Weights Selection

Assume that node A wants to start a new communication as the transmitter.

- If there are no CTSs overheard, then there is no other receiver in the collision domain who might get interference from node A. In this case, use default weight vector for transmission.
- Otherwise, let Φ be the set of nodes whose CTS has been overheard by this node. Solve the following set of equations to determine its weight vector w_A to use for transmission.

$$\forall I \in \Phi,\ \ w_A^T(Hw_I) = 0$$

where w_I is the weight vector used by node I.

3.2.2. Receiver Weights Selection

Consider a receiver B which has received an RTS from transmitter A. Transmitter A used weight vector w_A while transmitting. Node B needs to decide its weight vector w_B.

- If the intended transmitter is the only transmitter, then there will not be any other RTS it would have overheard. In this case, solve the equation $w_A^T(Hw_B) = 1$ to get the required weight vector w_B.
- Otherwise, search for the interfering transmitter in the same collision domain, i.e., the one whose *RTS* has been received and calculate its weight vector w_I.
- On receiving the *RTS* of the intended transmitter, calculate its weight vector w_T.
- Let Φ be the set of nodes whose RTSs have been heard earlier. Solve the following simultaneous equations to get w_B:

$$\forall I \in \Phi,\ \ w_I^T(Hw_B) = 0\ \ and$$

$$w_A^T(Hw_B) = 1.$$

Now, it is clear from the above discussion that by choosing the transmitter and receiver weights appropriately, we can have more number of simultaneous communications compared to IEEE 802.11 based MAC. With these modifications at the MAC layer, next section discusses the relevant changes required at the TCP layer.

4. TCP Enhancement with MIMO based MAC

There have been a lot of studies reported in the literature which show that TCP performance is very poor in Mobile Ad Hoc Networks (MANET) [3, 7]. In wireless networks, performance of TCP heavily depends on the design and operation of lower layer protocols. Specifically, the routing and MAC layer protocols can impact the performance of TCP in MANETs. For example, high collision rate in the MAC layer can degrade the performance of TCP significantly.

TCP congestion window limit (CWL) plays an important role in its performance. It has been reported that increasing the TCP CWL to a large value degrades its performance [4] in MANETs. Hence, there must be an upper bound on the TCP CWL. In [1], the authors have reported an upper bound on CWL for 802.11 based MANETs, which is approximately $1/5$ of the Round Trip Hop Count (RTHC). However, this upper bound may not be appropriate for MIMO based 802.11 network. In the following section we first describe the basic method in deriving the expression for upper bound on CWL and then derive the expression for tighter upper bound with the efficient MIMO based MAC (discussed in the previous section).

4.1. Computation of TCP Congestion Window Limit

Let us consider a chain topology of h hops in the forward path and m hops in the reverse path. We index the forward links by the hop count from the leftmost node. Thus, the forward links are identified as $l_1, l_2, .., l_h$. A link, l_i $(1 \leq i \leq h)$ has bandwidth b_i and c_i in the forward and reverse path respectively. The bottleneck bandwidth on the forward path is given by $B_{min} = \min_i b_i$ $(1 \leq i \leq h)$ and that on the reverse path is $C_{min} = \min_i c_i (1 \leq i \leq m)$. Let S be the TCP packet size or segment size. When the data packet travels from source to destination, the transmission delay on the forward path is given by

$$S/b_1 + S/b_2 + ... + S/b_h \leq S/B_{min} + S/B_{min} + ... + S/B_{min}$$

$$\leq h * (S/B_{min})$$

Similarly, on the reverse path, for the TCP ACK of size $X \leq S$, the delay is

$$X/c_1 + X/c_2 + ... + X/c_m \leq X/C_{min} + X/C_{min} + ... + X/C_{min}$$

$$\leq m * (X/C_{min}) \leq m * (S/C_{min})$$

Thus, the round trip delay (RTD) at the TCP layer is given by

$$RTD \leq h * (S/B_{min}) + m * (X/C_{min})$$

$$\leq h * (S/B_{min}) + m * (S/C_{min}) \qquad (1)$$

Assuming $B_{min} = C_{min}$, (1) reduces to

$$RTD \leq h * (S/B_{min}) + m * (S/B_{min}) \leq (h + m) * (S/B_{min}) \qquad (2)$$

TCP CWL depends on the Bandwidth Delay Product (BDP) of the path. The BDP of a link is defined as the product of the link bandwidth and round trip delay along the link. A TCP connection may span over multiple links. Hence, we define BDP of a TCP path as the product of the bottleneck bandwidth and the round trip delay along the path. Hence, CWL of TCP is given by

$$CWL \leq B_{min} * \ RTD$$

Using (2), we have

$$CWL \leq B_{min} \ * \ (h + m) * (S/B_{min})$$

$$CWL \leq (h + m) \ * \ S$$

which can also be expressed as

$$CWL = k * (h + m) \ * \ S, where\ k\ is\ a\ constant < 1 \qquad (3)$$

Thus, the TCP CWL depends on the product of the packet size and round trip hop count (RTHC).

4.2. TCP Congestion Window Limit for MIMO-based System

In the previous section, we found that the TCP CWL can be calculated from the product of the packet size and round trip hop count if the constant factor k is known. However, while calculating the above expression, we have implicitly assumed that the every link along the TCP path is filled up to the extent of bottleneck bandwidth of the path. This implies that transmissions can happen from each hop simultaneously. But this is not possible in wireless multihop network because of interference from neighboring nodes and due to hidden node and exposed node problems. As discussed in the previous section, MIMO-based systems can have maximum of $h/2$ packets in transit whereas in wireless system without MIMO there can be maximum of $h/3$ packets in transit for a given path. Thus, $k \leq 1/2$ for MIMO system and $k \leq 1/3$ for system without MIMO. But for TCP connections, there will be ACK packets in the reverse direction, which contends for the medium with the Data packets. This further reduces the number of simultaneous communication that can happen which means that k is further reduced. This can be explained by referring to the Figure 3. When ACK packets are accounted for, number of simultaneous communication can potentially reduce by half, which means k may reduce by half. Thus, for TCP connections in MIMO-based system $k \geq 1/4$, whereas for system without MIMO $k \geq 1/6$.

Hence, TCP CWL can be calculated in terms of the round trip hop count and packet size, i.e.,

$$CWL = k * (h + m) \ * \ S, \qquad (4)$$

where k is a constant given by

$$1/4 \leq k \leq 1/2,\ for\ MIMO\ system$$

$$1/6 \leq k \leq 1/3,\ for\ systems\ without\ MIMO$$

According to equation (4), we require round trip hop count to calculate CWL of TCP. This information can be made available by cross layer interaction, i.e., by letting the layers below TCP provide the hop count information. In wireless adhoc networks, round trip hop count can be easily calculated either by using source routing protocol like DSR, or by incorporating the mechanism of counting number of hops when data or ACK packets traverse a path. Once the round trip hop count is known, the only unknown left in equation (4) is k. Hence, to get the appropriate value of k, we performed the simulation experiments described in the next section. Using this value of k, CWL of TCP can be bounded, which should lead to better throughput.

5. Simulation Experiment

In this section, we first present the details of simulation setup we used to evaluate the performance of MIMO-based 802.11 system based on the theory presented in this paper. Then we discuss various results obtained using our simulator to validate our theory.

5.1. Simulation Setup

We have developed a simulator to evaluate the performance of MIMO-based wireless network proposed in this paper. The simulator is written in C++. We have assumed a constant packet size of 2312 bytes, which is the maximum size of packets in an 802.11 network and a chain topology in which each link bandwidth is set to 11 Mbps. All our simulation experiments were

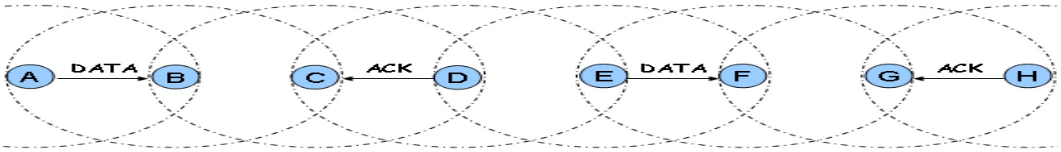

Figure 3: Illustration of Number of Simultaneous Transmissions Due to TCP Acknowledgement

run for 100,000 packets. The buffer size at each node is limited to 25 packets. Shadowing model used in our simulation is exactly same as the default configuration of shadowing model of ns2 [8]. Basically, ns2 shadowing model has two components. One component accounts for path loss and the other provides log normal variation in received power. Each node is assumed to have two antennas. For each transmit-receive antenna pair, the real and complex part of channel coefficient is modeled as a Gaussian random variable with mean zero and variance of 0.5 and the two random variables are independent of each other. For most of our experiments we have assumed that the transmission range of the nodes is same as the interference range. However, we also report some results corresponding to the case where interference range greater than transmission range and in such case, we mention this configuration.

5.2. Simulation Results

In all our experiments we created saturation condition at the TCP layer, i.e., TCP always has required number of packets to send whenever it receives an ACK so that the TCP window becomes full. In unsaturated condition, although congestion window size can grow to large value, TCP will not have enough packets to send at the rate corresponding to the congestion window size. Hence, TCP is run in saturation condition to test congestion window limit. It was found that TCP was running at saturation with offered load of 3 Mbps. Hence, all our experiments are run with 3 Mbps offered load.

To determine the effect of CWL on TCP throughput, TCP throughput was measured at different CWL values. Figure 4 shows CWL versus TCP throughput for different hop counts. As shown in Figure 4, for low values of CWL, when CWL increases, the TCP throughput also increases. But the throughput saturates after certain CWL value. Further, for larger hop count value, throughput saturates at a higher value of CWL. This is expected, since for larger hop count, more packets can be in transit (without affecting congestion) and hence CWL can be larger. So it is obvious that CWL should be set to the value which corresponds to the saturation throughput, so that TCP can maximize its throughput. From this observation, we can conclude that for each hop count there exists a CWL beyond which there is no significant increase in TCP throughput. We refer to this CWL as *optimal CWL*. Note that setting CWL to a value larger than optimal CWL would worsen the congestion in the network, without increasing the throughput significantly. This fact is obvious from Figure 5, where we measure the average end-to-end packet delay for two cases, one in which CWL is set to the optimal value and the other where there was no limit put on the congestion window size. It is clear that without any limit, the average delay of packet is higher than the case where CWL is set to the optimal value. This is because without any limit on CWL, CWL can become larger than the optimal CWL, which leads to congestion in the network. This, in turn, increases end-to-end delay.

As given by (4), TCP CWL is proportional to the RTHC and the proportionality constant k ranges from $1/4$ to $1/2$. To determine appropriate value of k, we obtained optimal CWL from Figure 4 for each hop count by noting down the CWL value which corresponds to saturation throughput. Figure 6 shows the variation in optimal CWL with Round Trip Hop Count (RTHC). The value of parameter k is calculated from this graphs as follows. For each RTHC we compute the corresponding value of k using (4) and then the final value of k is set to the average of these k's (obtained for different RTHC). Based on this, k is found to be $1/2.8$, when transmission and interference range are same. This k can be used to set the CWL of a TCP connection spanning any number of hops. We verify this fact by comparing CWL which is computed using $k = 1/2.8$ to the optimal CWL corresponding to saturation throughput (obtained from Figure 4). From Figure 8 it is clear that these two values are close to each other. Hence, $k = 1/2.8$ is the appropriate value of k that can be used to limit congestion window for any value of RTHC.

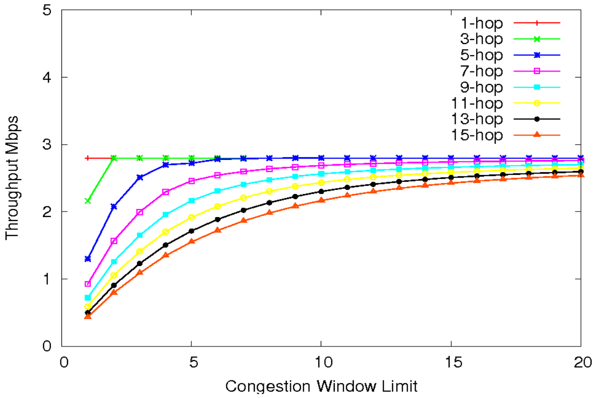

Figure 4: CWL vs. Throughput

In [1], the authors have calculated the value of k to be $1/5$ for 802.11 system without MIMO capability, but have assumed interference range to be larger than transmission range. We want to show that the same k value should not be used in MIMO based system, hence we configured our system to have interference range greater than transmission range and determined the value of k (using the same method as explained above) to be approximately $1/3$. Figure 7 plots TCP throughput obtained at optimal CWL with $k = 1/3$ and $k = 1/5$ versus RTHC. It is clear that with $k = 1/3$, the throughput is much better than that with $k = 1/5$. Hence, for MIMO-based system, TCP should use $k = 1/2.8$, when transmission range is equal to interference range, whereas $k = 1/3$ should be used when interference range is more than the transmission range.

6. Conclusion

Traditional MAC protocols for IEEE 802.11 network are not suitable for MIMO-based system because they lead to lower throughput. Hence, in this paper, we applied basic concepts of MIMO based MAC protocols to multihop 802.11 based wireless network. This modified MAC protocol allows more number of simultaneous communication and hence leads to higher throughput. Because of this modification at the MAC layer, TCP congestion window limit needs to be changed suitably to improve its performance. We presented a method to compute appropri-

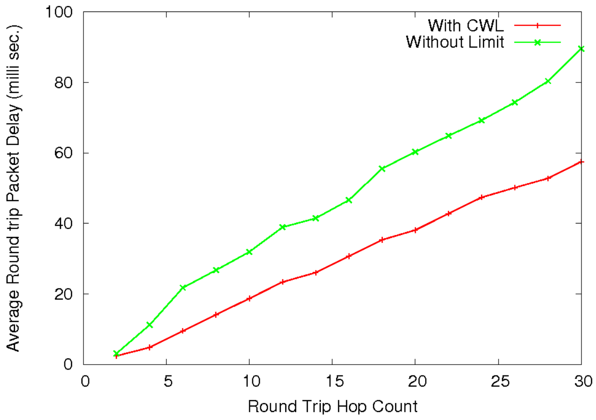

Figure 5: Round Trip Hop Count vs. Average Packet Delay

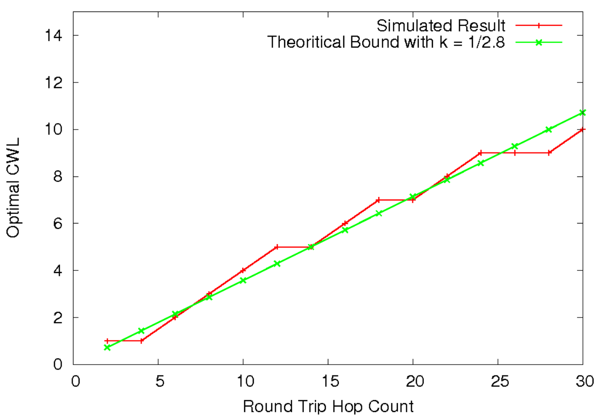

Figure 6: Round Trip Hop Count vs. Optimal CWL

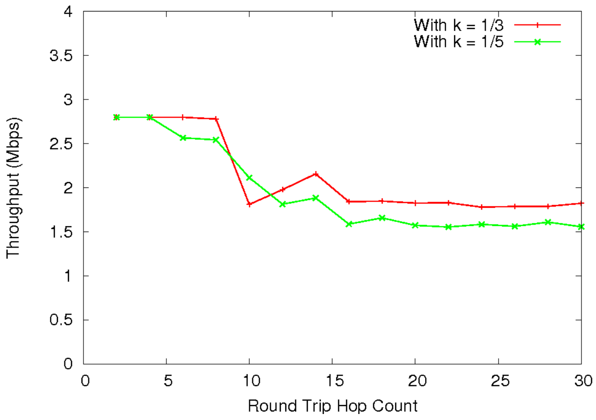

Figure 7: Round Trip Hop Count vs. Throughput

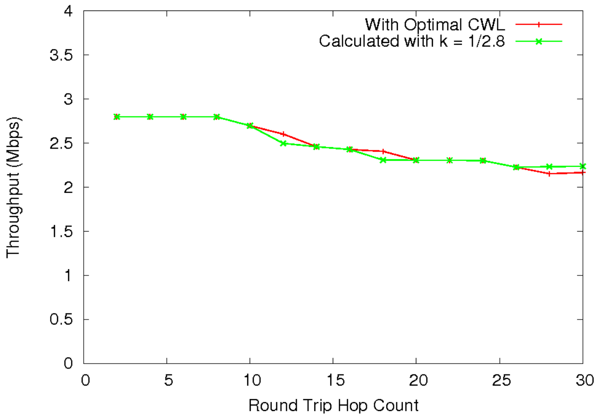

Figure 8: Comparison of Throughput using Optimal CWL and CWL Computed Using $k = 1/2.8$

ate value of the parameter k which is used to find the optimal CWL. We showed that using the optimal CWL, TCP has a better performance in terms of round trip delay. Our simulation data shows that when CWL can be empirically set to $1/2.8$ of round trip hop count to get TCP throughput close to that of optimal CWL. Hence, in a MIMO-based multihop network, the MAC protocol should be modified suitably to facilitate more number of parallel transmissions and correspondingly the TCP CWL should be set appropriately to improve the system performance.

References

[1] K. Chen, Y. Xue, S. H. Shah, and K. Nahrstedt. ”Understanding Bandwidth-Delay Product in Mobile Ad Hoc Networks”. *Computer Communications*, 27:923–934, 2004.

[2] D. J. Dechene, K. A. Meerja, A. Shami, and S. Primak. ”A Novel MIMO-Aware Distributed Media Access Control Scheme for IEEE 802.11 Wireless Local Area Networks”. *Local Computer Networks, 2007. LCN 2007. 32nd IEEE Conference on*, pages 125–132, 15-18 Oct. 2007.

[3] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla. ”The impact of multihop wireless channel on TCP throughput and loss”. *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, 3:1744–1753 vol.3, March-3 April 2003.

[4] M. Gerla, K. Tang, and R. Bagrodia. ”TCP Performance in Wireless Multi-hop Networks”. In *WMCSA ’99: Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*, page 41, Washington, DC, USA, 1999. IEEE Computer Society.

[5] K. Kim, P. Lorenz, and M. Lee. ”A new tuning maximum congestion window for improving TCP performance in MANET”. *Systems Communications, 2005. Proceedings*, pages 73–78, Aug. 2005.

[6] J. Mundarath, P. Ramanathan, and B. Van Veen. ”NULLHOC : a MAC protocol for adaptive antenna array based wireless ad hoc networks in multipath environments”. *Global Telecommunications Conference, 2004. GLOBECOM ’04. IEEE*, 5:2765–2769 Vol.5, 29 Nov.-3 Dec. 2004.

[7] K. Nahm, A. Helmy, and C.-C. J. Kuo. ”TCP over multihop 802.11 networks: issues and performance enhancement”. In *MobiHoc ’05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 277–287, New York, NY, USA, 2005. ACM.

[8] The Network Simulator ns-2 (v2.1b8a). http://www.isi.edu/nsnam/ns/, October 2001.

[9] J.-S. Park, A. Nandan, M. Gerla, and H. Lee. ”SPACE-MAC: enabling spatial reuse using MIMO channel-aware MAC”. *Communications, 2005. ICC 2005. 2005 IEEE International Conference on*, 5:3642–3646 Vol. 5, 16-20 May 2005.

[10] K. Sundaresan, R. Sivakumar, M. Ingram, and T.-Y. Chang. ”A fair medium access control protocol for ad-hoc networks with MIMO links”. *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, 4:2559–2570 vol.4, March 2004.

[11] A. Toledo and X. Wang. ”TCP performance over wireless MIMO channels with ARQ and packet combining”. *Mobile Computing, IEEE Transactions on*, 5(3):208–223, March 2006.

[12] H. Zhai, X. Chen, and Y. Fang. ”Rate-based transport control for mobile ad hoc networks”. *Wireless Communications and Networking Conference, 2005 IEEE*, 4:2264–2269 Vol. 4, March 2005.