Journal of Theoretical and Applied Information Technology
31st January 2018. Vol.96. No 2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DIMINISHING MAC OVERHEAD BY USE OF PERIODIC AND COOPERATIVE ACK SCHEMES IN 802.11 WIRELESS NETWORKS

[1]AYINEBYONA ELIAB, [2]JONGSUN PARK AND JUNGMIN SO*
Smart Computing Laboratory, Dept. of Computer Engineering, Hallym University
Chuncheon, South Korea
E-mail: eliabic@hallym.ac.kr, jongs007@hallym.ac.kr, jso1@hallym.ac.kr
*Corresponding author: Jungmin So

ABSTRACT

This paper presents schemes that reduces ACK (acknowledgement) overhead in IEEE 802.11 wireless networks and further improve network performance in high error rate cases. While the ACK takes only 20us in 802.11a at 54Mbps, a combination of SIFS period and preamble makes ACK the second largest overhead next to channel access overhead. To reduce ACK overhead, we proposed periodic and cooperative ACK schemes. In periodic ACK scheme, a single ACK is used to periodically acknowledge multiple packets received independently at different transmission occasions on each channel separately. However, the periodic ACK scheme suffers throughput degradation in high error rates. To overcome this issue, a cooperative ACK scheme was applied. In cooperative ACK, a single ACK is sent on one of the active channels to acknowledge multiple packets received on different channels collectively, rather than acknowledging multiple packets received on each channel separately. This reduces ACK timeouts that would result in excessive retransmissions since an ACK timeout on one channel would be recovered on another channel thus improving network performance. Extensive simulation in ns3 shows that the proposed schemes improve network throughput by considerable amount in various environments.

Keywords: *IEEE 802.11, MAC overhead, Multi-radio, Acknowledgment*

1. INTRODUCTION

The Wi-Fi physical layer data rates have rapidly scaled from 1 Mbps in traditional 802.11 to more than 1 Gbps in 802.11ac. This has significantly decreased data frame transmission time but posed a challenge of MAC (Media Access Control) overheads that stay constant at all channel bit rates. For instance, while data sending time decreases at higher rates the ratio of MAC overhead increases. This is largely due the fact that MAC overhead is independent of channel bit rates [1].

One way of reducing alarming MAC overhead in wireless network with high data rates is to use large packet size [10, 11]. For this reason, the 802.11n/ac standards adapted MAC-level frame aggregation techniques that were initially proposed in 802.11e [3]. Both Aggregate MAC Service Data Unit (AMSDU) and Aggregate MAC Protocol Data Unit (AMPDU) group several data frames into one large frame that can be transmitted in a single channel access. By using AMSDU and AMPDU, data transmission time can be increased in greater proportional to overheads hence allowing resource efficiency. Then instead of transmitting individual ACK for multiple packets in an A-MPDU, multiple MPDU are acknowledged in a single block ACK. However, it is not possible to always achieve large packet in practice. For instance, in [4], it was noted that 55% of internet packets are less than 100 bytes. Moreover, packet aggregation techniques are not applicable to delay sensitive application such as short HTTP transfers or remote desktop connection. This can also cause starvation of other nodes on the same channel.

In [1], a system to reduce channel access overhead using multi-radio approach was proposed. In their scheme, nodes do contention on one channel and send packets on other channels without contention. It is assumed that a node occupying contention channel can directly send data on other channels, without contention. Magistretti et al. [6] presented WiFi-Nano, a system that reduces slot time from 9us to 800ns and uses speculated ACK to remove inter-frame spacing, thereby reducing random backoff time and acknowledgment overhead. Channel access overhead has also been dealt with in [7, 8]. However, acknowledgment overhead has generally received less attention in the research community even though it is the second largest overhead next to channel access. In this paper we propose periodic and cooperative ACK schemes. Both schemes reduce ACK overhead. Our proposed schemes can be adapted by channel access overhead reduction systems like the one presented in [1] in order to increase benefits.

The periodic ACK scheme aims at reducing acknowledgement overhead discussed in Section 2. The ACK itself takes 20µs in 802.11a at 54Mbps, however its overhead is largely influenced by the preamble and turnaround time. We include an ACK flag in the MAC header. The sender sets the flag to either 1 or 0 to mean ACK required or not required respectively. To diminish this bottleneck, a node sends multiple packets with ACK flag set to 0 followed by ACK flag set to one packet. Upon reception of a packet with ACK required flag raised, the receiver sends back a single ACK to acknowledge all independently received packets. The ACK contains a bitmap that bears information of all received packets. The term periodic emanates from the fact of sending one acknowledgement every after N received packets. The co-operative ACK scheme is an extension of periodic ACK in multi-radio nodes operating on orthogonal channels. Instead of sender receiving ACK on each channel a single ACK is used to acknowledge multiple packets received on different channel.

In this work, the key contributions are summarized as follows:

A) Periodic ACK scheme embraced both single and multiple radio nodes. In periodic ACK, nodes use a single ACK to acknowledge multiple packets.

B) Co-operative ACK works on top of periodic ACK and adds a feature that allows multi-radio nodes to send a single ACK on one channel to acknowledge all packets received on all channels.

C) The scheme required exploitation of MAC header least used fields in order to include ACK flag/ no ACK flag in every data packet. Acknowledgment packet format was modified to enable multiple packet acknowledgment in a single ACK packet using a bitmap.

D) In one ACK per received packet scenario the sender queue keeps the sent packet before enqueuing of another packet from upper layer. It can be dropped in case successful transmission or retransmitted in case of packet loss. However, in periodic ACK multiple packets can be acknowledged by a single ACK. Retransmit queue was introduced to keep track of transmitted packets waiting for acknowledgement.

The rest of this paper is organized as follows, in Section 2, we discuss some related works, and the differences between our proposal with them . In Section 3, we analyze MAC overheads through calculations. In Section 4, we present in details our proposed scheme. In Section 5, under various configurations and conditions, we evaluate performance of periodic and cooperative schemes. We conclude the paper in Section 6.

2. RELATED WORK

IEEE 802.11 wireless LANs supports many data rates by applying multiple sets of modulation and channel coding schemes [13]. However, the transmission time for control frame and inter-frame spaces are always fixed. According to the authors in [14], an upper bound for throughput and performance cannot be gained by simply increasing data rates without reducing the overheads.

In IEEE 802.11e a block ACK was presented to reduce overheard [15]. A transmitter is assigned a fixed time called transmission opportunity (TXOP), in which a block of data frames is transmitted. Subsequently, the receiver aggregates the multiple ACK frames and sends it as a single BA that simultaneously acknowledges the status of all transmitted data frames by using a bitmap.

In IEEE 802.11n, the throughput enhancement is achieved by aggregating multiple packets before transmission [16]. Aggregation has several advantages, such as reducing the channel waiting time during the backoff process for transmitting consecutive frames. Moreover, the period used for preambles, inter-frame spacing and header transmission is reserved. While aggregation of multiple frames can improve the network throughput in error free environment, large data packets may lead to starvation of other stations on the same channel, under erroneous channel conditions. Corruption of a large aggregated frame would waste substantial channel time, in which results in degrading MAC efficiency. Therefore, frame aggregation mechanism can not be considered as a good candidate for wireless network, which is error-prone due to its dynamic traffic nature.

In [12], the ACK have been exploited to reduce backoff time, packet collisions and explicit ACK frame. The ACK is piggybacked on the data frame and each ongoing transmission passes token to another host who will then become a high priority candidate for the next transmission period through overhearing mechanism. However, this system requires much power consumption due overhearing. The scheme proposed in this paper does not require overhearing.

In efforts to reduce ACK overheard, Choudhury et al. [17] proposed an “implicit MAC acknowledgement scheme” where the explicit ACK frame has been eliminated by piggybacking ACK information in RTS/CTS frame. However, the RTS/CTS control frames precedes each data frame which further increases overhead. In contrast, for our scheme, RTS/CTS have nothing to do with ACK, the ACK is used periodically and the use of RTS/CTS remains optional. Additionally, to be able to use implicit-ACK the scheme in [17] requires a sender to have at least two packets destined for the same receiver.

Proposed schemes perform well in different environments and do not pose conditions on packet sizes like block ACK overhead reduction system. Our schemes do not require elimination of control packets in IEEE 802.11 wireless networks. In case of Multi-radio nodes ACK overhead can only be maintained on one channel while keeping other channels free of ACK overhead.

3. THE IEEE 802.11 DISTRIBUTED COORDINATE FUNCTION (DCF)

The IEEE 802.11 DCF allows a node to reserve the channel for data transmission by exchanging RTS/CTS with the target node. A node wishing to transmit data packet to another node, first sends an RTS (Ready to send) to the destination. The destination station acknowledges the reception of RTS frame by sending back a CTS (clear to send) packet to the sender. RTS and CTS frame include the period of time for which the channel will be in use. Other hosts that overhear these packets must differ their transmission for the time slot specified in the packets. Basically, each node maintain a variable termed as network allocation vector (NAV) that keeps truck of the period of time it must differ transmission. The whole of this process is referred to as Virtual Carrier Sensing, which allows the area around sender and receiver to be reserved for communication, thus avoiding the hidden terminal problem [18]

Figure 1 illustrates the mechanism of IEEE DCF. When node X is transmitting a packet, node W overhears the RTS packet and sets its NAV until the end of ACK, and node Z overhears the CTS packet and sets its NAV until the end of the ACK. Stations contend for the channel after the completion of transmission. In this setup, node X is a hidden terminal to node Z that would led to collisions at Y, if Z attempted transmitting to Y while X is transmitting to Y. However, such collisions are eliminated by virtual carrier sensing.

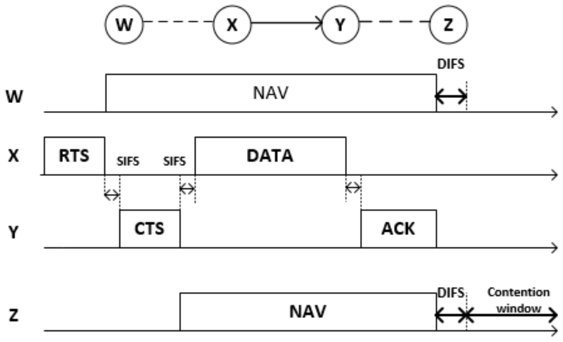


Figure 1: Ieee 802.11 Dcf Mechanism

If a node has a packet to transmit while the channel is busy, it performs a random backoff by choosing a backoff counter. For each slot time interval, during which the medium stays idle, the random backoff counter is decremented and it freezes the countdown when channel becomes busy again. When the backoff reaches counter zero, the node reserves channel by sending an RTS to the target node. It’s possible for two nodes to select the same backoff counter leading to the loss of RTS due to collision. Since the probability of collisions increases as the number of node increases, a sender would interpret the absence of a CTS as a sign of congestion and react by doubling its congestion window to lower chances of another collision.
Prior to sending a packet when the channel becomes idle a node has to wait for interframe spacing. A node waits for a DIFS before sending an RTS, but waits for a SIFS before sending a CTS or an ACK. This allows the ACK to win the channel when contending with RTS or DATA packets since SIFS duration is smaller than DIFS. Note that in our experiments CTS/RTS where turned off.

4. MAC OVERHEAD ANALYSIS

In this section we analyze MAC overheads present in 802.11 with RTS/CTS turned off. Considering 802.11a at a maximum data rate of 54 Mbps, each OFDM symbol carries 216 data bits that are spread out over 48 subcarriers. Assuming a UDP payload of 11648 bits and a symbol duration of 4µs, only 216µs would be used to transmit the packet but this duration is confined by overheads, as in Figure. 2.

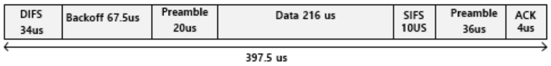
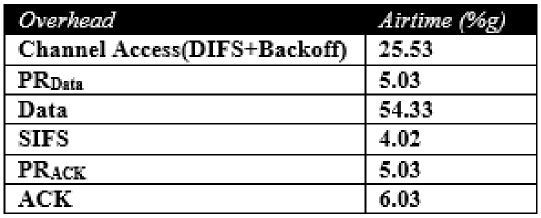


Figure 2: 802.11a Packet Transaction Time At 54 Mbps

Before every packet transmission DCF requires a node to go through DIFS which is 34µs, followed by 67.5µs average random backoff period to ensure fair medium sharing. It then sends a preamble and PLCP header which takes 20µs in 802.11a, then sends data followed by SIFS of 16µs, upon packet reception the receiver sends back ACK which also require preamble and PLCP header that takes total time of 24µs.

Table: MAC Airtime Parameters



Overhead	*Airtime (%g)*
Channel Access(DIFS+Backoff)	**25.53**
PR_{Data}	**5.03**
Data	**54.33**
SIFS	**4.02**
PR_{ACK}	**5.03**
ACK	**6.03**

The channel access (DIFS and Backoff) mechanism has the largest portion of airtime utilization of up 25.53% of the total time, as in Table 1. In 802.11a the slot time of 9µs is used and thus DIFS and average backoff are calculated from $SIFS + 2 \times slottime$ and $(C_{min}/2) \times slottime$ respectively. The 9µs slot time is required for packet detection and turnaround period whereas the backoff time is used to alleviate collisions and allow fair medium access as aforementioned.

Preamble also takes a considerable large duration as seen from Table 1. There are two preamble periods PR_{Data} and PR_{ACK} which are preamble periods for data and acknowledgment respectively each with 5.03% which adds up to 10.06%. The goal of preamble is to facilitate receiver in timing synchronization, carrier-offset recovery and channel estimation. Its duration has continuously increased in new standards for instance from 20µs in 802.11a/g to 36µs in 802.11n to aid communication of incumbent and newer standards.

802.11 standard dictates SIFS period of 16µs followed by the Ack. Therefore, ACK overhead can be recalculated as $Ack_{Overhead} = SIFS + PR_{Ack} + Ack$, which is 15.08 % of packet transmission time, see Table 1. Transmitting UDP load of 11648 bits in the time frame shown in Figure 2 translates into throughput of 29.30Mbps with all overheads and 34.09Mbps without ACK overhead. In brief ACK causes 5Mbps loss.

Basing on calculation of ACK overhead it's the second largest overhead next to channel access overhead that has gained popularity among researchers. We therefore focus on reducing ACK overhead by sending a single ACK for multiple independently received packets on multi-channel or single channel.

5. PROPOSED ACK SCHEMES

The periodic and cooperative ACK schemes have been implemented at the MAC layer to support both single and multi-radio nodes. We use the term periodic ACK to mean sending one ACK to acknowledge multiple packets that have been sent and received independently whereas cooperative ACK refers to sending an ACK on one channel to acknowledge packets received on other channels. Periodic ACK is fundamentally different from block ACK scheme where multiple packets are aggregated into a single A-MSDU/A-MPDU and transmitted as one large packet in which the block ACK is used to acknowledge individual packets concatenated in a single A-MSDU/A-MPDU.

5.1. Periodic ACK in Single Radio

The IEEE 802.11 standard dictates sending ACK for every packet reception, as in Figure 3. However, this incurs 5Mbps loss as discussed in in Section 2.

Figure 3: C Sends ACK for Every Data Received from S

To overcome loss due to ACK overhead we adapted a scheme of sending one ACK to acknowledge multiple packets as in Figure 4.

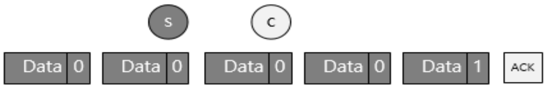


Figure 4: C Sends A Single ACK For Multiple Data Packet Received Independently From S

. In Figure 4, 0 means that data being sent does not require an ACK whereas 1 implies that data being sent require an immediate ACK that will acknowledge all receptions. To implement this procedure, we exploited MAC header fields that are rarely used. In MAC header we used order field which is a sub-field of frame control. The number of packets sent before receiving the ACK can be valid from one to N number of packets, where N is an integer. At the receiver side, the receiver process every packet received to judge if it's an ACK or No ACK required packet and respond accordingly.

5.2. Periodic ACK on Multi-radio Nodes

Multi-radio nodes are preferred in today's wireless network due to their performance benefit compared to single radio nodes. In this section we detail extension of periodic ACK from single radio to multiple radio nodes. Note that multi-radio nodes are capable of using N interfaces. The first technique applied is straightforward, we send multiple packets on each channel and receive an ACK on every channel separately. In Figure 5, two nodes S and C are equipped with two radios operating on orthogonal channels and S can receive two ACKs from different channels during ACK period.

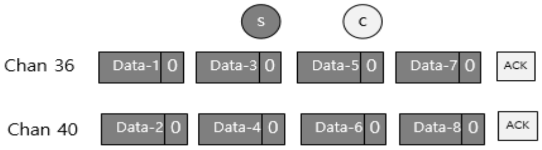


Figure 5: *S sends 3 non-ACK packets on each channel and request for an ACK to acknowledge all packets on fourth packet on each channel.*

However, wireless channel medium is a victim to dynamic interference which would led to great variations among channels depending on interference levels. Assuming that channel 36 has the highest interference level compared to channel 40, for instance after S sent third packet sequenced number 5 on channel 36, as in Figure 5. The channel was occupied by another pair of nodes and S has to backoff and waits for the second chance to win transmission time slot. On the other hand, node C received all packets on channel 40 and sent back acknowledgment packet on channel 40 only hence node S will keep waiting for the ACK on channel 36. Or assume that packets got lost due to collision and S has to retransmit all packets on channel 36.

To overcome this, we proposed cooperative ACK where a single ACK can acknowledge packets received on all channels and reset ACK period in the parallel channels. For instance, if a node received a packet with ACK demanding flag raised on channel 36 but waiting for packet with ACK demanding flag raised on channel 40. We send an ACK acknowledging all packets received by node C on channel 36 and then reset ACK period of channel 40. Impacts of lost ACKs that trigger retransmission of all packets in the retransmit queue with sequence number higher than last acknowledged packet would be reduced and improve performance in networks with high levels of interference. This approach would further be beneficial in a situation where one channel uses lower ACK period whereas the other channel uses higher ACK period say n and 2n respectively. The channel with lower period would always send an acknowledgment every after n packets to acknowledge all packets received on all channels and reset the period. If the process goes on for a while, it would result in sending ACK on one channel hence removing ACK overhead on the other channel completely.

5.3. Packet Loss and Retransmit Queue

Packets are distributed on participating interfaces in one to one ratio similar to round robin, which is the default bonding mode [9]. At the sender side we created a retransmit queue to temporarily store all sent packets. The sender maintains sender queue and retransmit queue. Sender queue keeps all packets from the upper layer to be transmitted whereas retransmit queue temporally stores transmitted packets. Received ACKs are compared with all packets in retransmit queue. Packets in the transmit queue with a sequence number higher than thesequence of the last acknowledged packets are considered lost and should be transmitted. The sender switches to either of queues depending on task to be performed but transmit queue is a given a priority.

6. EVALUATIONS

All experiments were operated on 20 MHz channel using IEEE 802.11a at 54Mbps, where ns3 was used as simulator. Results of experiments are presented in the rest of this section..

6.1 Varying ACK Period in Single Radio

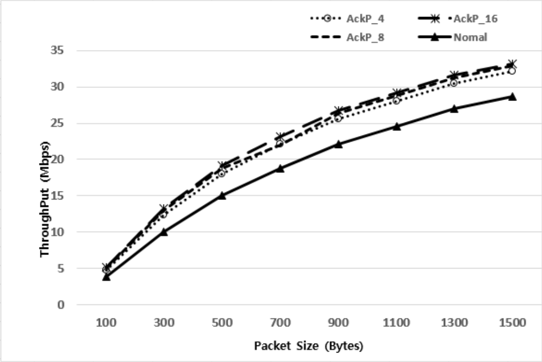


Figure 6: UDP throughput for single radio using various ACK period and normal

In the first experiment, we varied ACK period and measured throughput of UDP traffic to study the benefit of using periodic ACK compared to normal 802.11. The ACK period of 4, 8 and 16 are used and compared with normal case. Packet sizes vary from 100 to 1500 bytes. Figure. 6, shows that periodic ACK improves throughput up to 15%. The throughput while using different ACK intervals and normal are high when payload is large packet due to generally decreased MAC overhead ratio and the reverse is true. The performance difference of different ACK periods is minimal due the small value of throughput gain for different intervals.

6.2 Impact of Error Rate with and without ACK period using large traffic load

In this experiment, we fixed packet size to 1500 bytes and varied error rate to validate impacts of error rate on large packets when using the periodic ACK in single radio environment.

In Figure 7, at zero loss rate using large packet size offers higher throughput than using small packet, see Figure 8 due to less overhead. The largest periodic ACK offers 17% throughput gain compared to normal. However, as loss rate increases, the network performance degrades generally, but most significantly when using higher ACK period. This is due to the large number of retransmissions that require long transmission time as the packet size is large. Basing on the results, it's better to use normal ACK when the error rate is higher than 20%.

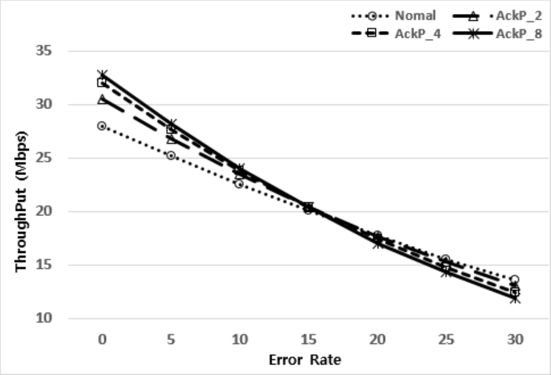


Figure 7: Performance With And Without Periodic ACK At Different Loss Rate

6.3 Impact of error rate with and without ACK period using small traffic load

In this experiment, we fixed packet size to 500 bytes and varied rate to validate impacts of error rate on small packets when using periodic ACK scheme on single radio.

In Figure 8, a combination of small packet and periodic ACK of 8 improves network performance up to 30%, which is 13% higher than using large packet since when sending small packets the ratio of payload to overhead is small, so removing overhead offer considerable gains. The overall performance is lower than using large packet size, this is due to the fact that large packet size decreases the ratio of overhead. Periodic ACK offers higher performance gain of up to 25% since transmission time reduces and time required for retransmission is lower than when using large packet. It's good to use small packets in networks with high loss rate.

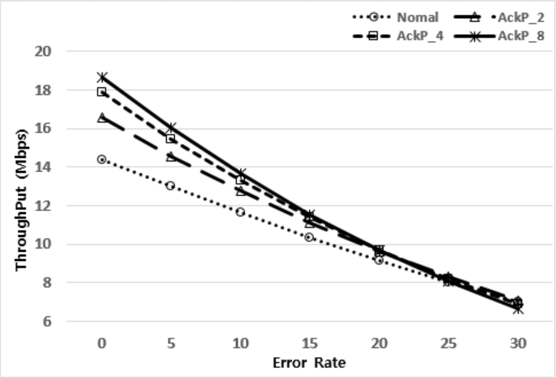


Figure 8: Performance With And Without Periodic ACK At Different Loss Rate Using Payload Of 500 Byte

6.4 Impact of Cooperative Ack

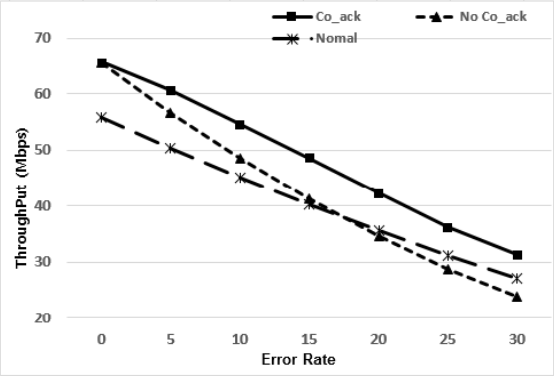


Figure 9: Performance Of Cooperative Ack, Non Cooperative And Normal At Various Loss Rate.

Cooperative ACK is applied in multi-radio nodes capable of operating on orthogonal channels. Nodes send one ACK on one channel to acknowledge packets received on other channels and reset ACK transmission period when cooperative ACK is on (Co_ack), send one ACK on every channel to acknowledge packets received on each channel independently at every ACK period (No Co_ack) and Normal when nodes send ACK for each and every packet received on each channel independently.
In this experiment we used ACK period of 8, two radios and a constant payload of 1500 bytes

In Figure 5, at 0% loss rates both cooperative ACK and no-cooperative ACK outperforms normal by 17% due to benefits of periodic Ack. As error rate increases beyond 15% normal performs better than no-cooperative due long ACK period of 8 that results in excess retransmission in cases of ACK loss whereas only one packet would be retransmitted in normal setup. At 30% error rate, cooperative ACK improves by 30% and 15% compared to no-cooperative and normal respectively. Cooperative ACK continuously performs better than other setup since pitfalls of lost ACK on one channel that would require retransmission are recovered from by receiving ACK on another channel acknowledging all packets hence minimizing the number of retransmissions.

6.5 **Impact of Cooperative ACK using four radios.**

To understand the impact of cooperative ACK in networks with high level of interference, we increased the number of radios to four and kept as parameters same as in Section 4.3.

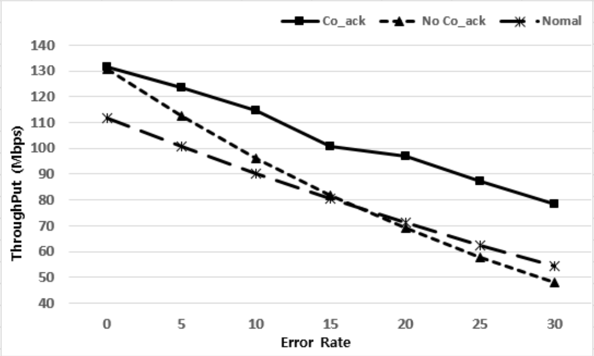


Figure 10: Performance Of Cooperative Ack, No Cooperative And Normal At Various Loss Rate Using 4 Radios

In Figure 10, at 0% the performance gains are similar to the ones discussed in Section 4.3. However, the gains of cooperative ACK are significant in high loss rate, at 30% error rate cooperative ACK outperforms both no cooperative and normal by 63% and 42% respectively. It's better to use of collaborative ACK technique and many radios in high error rates conditions

7. CONCLUSION AND FUTURE WORK

MAC overhead and collisions are the major sources of throughput degradation in IEEE 802.11 wireless network. Using periodic scheme reduces ACK overhead which is the second largest MAC overhead and improve network efficiency. Cooperative ACK was introduced to further reduce MAC overhead and improves performance in high error rate networks. Potential effectiveness of our schemes was proved through ns-3 simulations, cooperative ACK improves performance by more than 63% in high error rates cases.
In the future, we would like to test our schemes in testbed and combine the proposed schemes with other channel access reduction system both in simulations and test-bed.

ACKNOLEDGEMENT

This work was supported in part by the National Research Foundation of Korea (NRF-2016R1D1A1B03930289) and Hallym University Research Fund (HRF-201610-009)

REFERENCES

[1] J. So, C. Yoon, J. Lee, J. Lee, and Y. Ko, "Mac Overhead using Multiple radios in IEEE 802.11 based Wireless Networks," *International journal of control and automation Animation* Vol 6, No. 6, pp. 267-276, 2013.

[2] X. Yang and N. Vaidya, "Spatial Backoff Contention Resolution for Wireless Networks", *UIUC Technical Report*, 2006.

[3] IEEE, Part 11: Wireless medium access control (MAC) and physical layer (PHY) specifications: Amendment 8: Medium access control (MAC) quality of service enhancements. IEEE 802.11e, Sep, 2005.

[4] V. Raman and N. Vaidya, "WiSP: A Protocol for Overcoming MAC Overheads Using Packet Size Dependent Channel Widths", *in Proc. of IEEE SECON*, 2011.

[5] M. Bauer, B. Bruegge, G. Klinker et al., "Design of a component-based augmented reality framework," in *Proc. of International Symposium on Augmented Reality (ISAR)*, 2001.

[6] E. Magistretti, K. Chintalapudi, B. Radunovic and R. Ramjee, "WiFi-Nano: Reclaiming WiFi Efficiency Through 800ns Slots", in Proc. of ACM MobiCom, 2011.

[7] S. Sen, R. Choudhury and S. Nelakuditi, "No Time to Countdown: Migrating Backoff to the Frequency Domain", in Proc. of ACM MobiCom, 2011.

[8] K. Chintalapudi, B. Radunovic, V. Balan and M. Buettener, "WiFi-NC: WiFi Over Narrow Channels", in Proc. Of USENIX NSDI, 2012.

[9] J. So, J. Park, and A. Eliab, "Wireless Interface bonding supporting In-order Delivery and Automatic Load Balancing, TVT.2016.2612245, IEEE Transaction on Vehicular Technology.

[10] B. Ginzburg and A. Kesselman, "Performance Analysis of A-MPDU and A-MSDU Aggregation in IEEE 802.11n", in Proc. of IEEE Sarnoff Symposium, 2007.

[11] Y. Lin and V. Wong, "Frame Aggregation and Optimal Frame Size Adaptation for IEEE 802.11n WLANs", in Proc. of IEEE

[12] Shegufta B. Ahsan and Nitin H. Vaidya, O-ACK: An Adaptive Wireless MAC Protocol Exploiting Opportunistic Token-Passing and Ack Piggybacking. Short paper at the 40th Annual IEEE Conference on Local Computer Networks (LCN), October 2015

[13] Tinnirello, I.; Choi, S. Efficiency analysis of burst transmissions with block ACK in contention-based 802.11e WLANs. IEEE ICC 2005, 5, 3455–3460.m

[14] Xiao, Y.; Rosdahl, J. Throughput and delay limits of IEEE 802.11. Commun. Lett. 2002, 6, 355–357

[15] IEEE Std 802.11e-2005. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY)

[16] IEEE Std., 802.11n-2009, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput, 2009.

[17] R. R. Choudhury, A. Chakravarty, and T. Ueda. Implicit mac acknowledgment: An improvement to 802.11. IEEE/ACM Wireless Telecommunications Symposium (WTS), April 2005

[18] F. A. Tobagi and L. Kleinrock, "Packet Switching in Radio Channels: Part II - the hidden terminal problem in carrier sense multiple-access modes and the busy tone solution," IEEE Transactions on Communications, COM-23, 1975.