

MAC Overhead Reduction using Multiple Radios in IEEE 802.11-based Wireless Networks

Jungmin So, Cheal-Hwan Yoon, Joo-Yub Lee, Jeong-Gun Lee and Young Woong Ko

Dept. of Computer Engineering, Hallym University

jso1@hallym.ac.kr

Abstract

In this paper, we propose techniques that reduce the MAC overhead in IEEE 802.11-based wireless networks. As the channel bit rate increases, MAC overhead becomes a major source of throughput degradation. This is due to the fact that the MAC overhead is rate-independent; it is fixed regardless of the data rate. Rate-independent overhead can be reduced by dividing a high-rate wideband channel into multiple low-rate narrow channels, and send data on these narrow channels simultaneously using multiple or compound radios. In this environment with multiple narrow channels, we propose two techniques that further reduce MAC overhead and thus improve resource efficiency. First, channel contention is done only on one of the narrow channels, called the contention channel. The other channels are accessed based on the status of the contention channel, without contention. This technique significantly improves network throughput, but has a problem in which if a collision occurs on the contention channel, collisions can also occur on other channels. To avoid this phenomenon, we propose the second technique in which a node observing collision notifies the senders that cause collision, so that they do not access other channels simultaneously. Extensive simulation results show that the proposed techniques improve network throughput by a considerable amount.

Keywords: IEEE 802.11, wireless LANs, multiple radios, MAC overhead, narrow channel

1. Introduction

As the channel bit rate increases, the MAC (Medium Access Control) overhead becomes a significant factor limiting the network throughput of IEEE 802.11-based wireless networks. The MAC overhead is necessary to avoid collision among multiple contending nodes. When the channel is sensed to be idle, a node has to wait for a time interval called IFS (Inter-Frame Spacing) followed by a random backoff period, before transmitting a packet [1]. If a collision occurs, the nodes increase their random backoff period in order to avoid collision in the next try. The problem with the MAC overhead is that they are independent of the channel bit rate [9]. As the channel bit rate increases, transmission time of a packet will decrease, assuming the packet size is fixed. However, the MAC overhead stays the same. As a result, the portion of MAC overhead when transmitting a packet increases with the channel bit rate, damaging the network throughput.

The simplest method for reducing the negative effect of MAC overhead is to increase the packet size [2, 3]. As shown in Section 2, the efficiency is significantly increased just by using a large packet instead of a small packet. For this purpose, the IEEE 802.11n standard includes two MAC-level data aggregation mechanisms; A-MSDU (Aggregate MAC Service Data Unit) and A-MPDU (Aggregate MAC Protocol Data Unit). Both mechanisms aggregate multiple packets into a single packet, thereby increasing packet size and resource efficiency.

A-MSDU and A-MPDU differ in when the aggregation occurs; A-MSDU aggregates packets before adding the MAC header, while A-MPDU aggregates packets after MAC header is added. Generally, A-MPDU offers a higher MAC throughput compared to A-MSDU. However, it is not always possible to use large packets. There are applications that use small packets that need to be sent as fast as possible, such as remote access applications or VOIP applications. It is reported that 55% of the packets in the Internet were less than 100 bytes in 2008 [4]. These small packets include TCP ACKs, which may degrade the connection throughput when delayed. Thus, packet aggregation is not the Holy Grail in overcoming MAC overhead.

Other approaches for reducing MAC overhead have been proposed as well. Sen *et al.*, [6] have proposed a mechanism that reduces time consumed in random backoff. In their proposed protocol Back2F, each node chooses a random integer which is mapped to a subcarrier in the frequency domain. Instead of doing random backoff in time, each node transmits a signal on the chosen subcarrier using a second antenna. The node that chose the highest integer wins the channel. This protocol can significantly reduce the MAC overhead, but requires a dedicated antenna for sending and receiving control signals. Magistretti *et al.*, [5] proposed WiFi-Nano, a system that uses 800ns slots, while the conventional IEEE 802.11n system uses 9 μ s slots. Since the random backoff time is based on the slot time, short slot times can significantly reduce MAC overhead and increase efficiency. Chintalapudi *et al.*, [7] proposed a system called Wifi-NC, a Wifi system that uses multiple narrow channels. The authors of [7] have developed a compound radio, which is a single wideband radio that combines multiple narrow channels, and is capable of transmitting, receiving and carrier sensing on the narrow channels independently. The techniques proposed in this paper can be implemented on top of this compound radio, as well as multiple conventional radios. Raman *et al.*, [4] proposed WiSP, a protocol that chooses the channel widths based on packet sizes. Based on the percentage of small and large packets, the channel is partitioned into a narrow and a wide channel, so that packets are transmitted on each sub-channel according to their sizes.

In this paper, we assume a narrow channel environment, such as one described in [7]. Instead of having each narrow channel follow the normal IEEE 802.11 DCF (Distributed Coordination Function), we choose one of the narrow channels as the *contention channel*, where nodes contend as in IEEE 802.11 DCF for the channel. In other channels, nodes get access to the channel based on the status of the contention channel. This technique further improves efficiency, since the contention overhead is reduced from a wideband channel to a narrow channel. Other frequency bands previously used for contention are now used for data transmission.

In Section 2, we analyze the efficiency of IEEE 802.11n under various environments and motivate our problem. In Section 3, we describe the proposed techniques in detail. In Section 4, the performance of the proposed techniques is evaluated using simulation. Finally, Section 5 concludes the paper with plans for our future work.

2. Efficiency Analysis of IEEE 802.11n

IEEE 802.11n can achieve 300Mbps of channel bit rate using a 40MHz channel, 64QAM modulation, 5/6 coding rate and 2 \times 2 MIMO (600Mbps can be achieved with 4 \times 4 MIMO). To find out the effective throughput, we have run a simple simulation using the ns-2 simulator [12]. Two nodes A and B are placed close to each other, and node A sends a high-rate CBR (Constant Bit Rate) flow to node B. The parameters in the simulations are set in accordance to the IEEE 802.11n standard. The slot time is 9 μ s, SIFS (Short Inter-Frame Spacing) is 10 μ s, and DIFS is 28 μ s. The minimum contention window CW_{Min} is 15. We assume 2 \times 2 MIMO,

and the Mixed mode which assumes all nodes in the system are 802.11n-compatible nodes. Thus, the time to transmit preamble and the PLCP header is $32\mu\text{s}$ [1]. The packet size is varied to see its impact on the throughput.

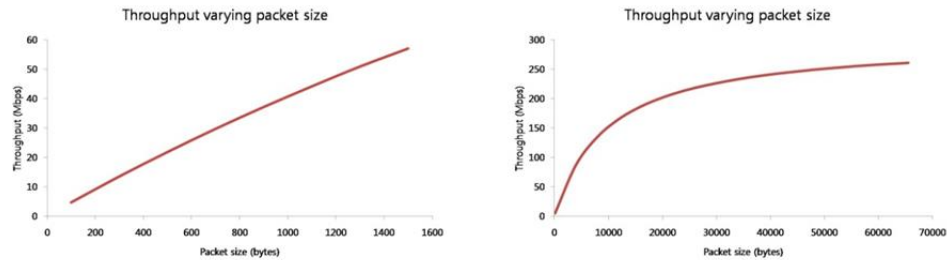


Figure 1. Effective throughput of IEEE 802.11n under various packet sizes

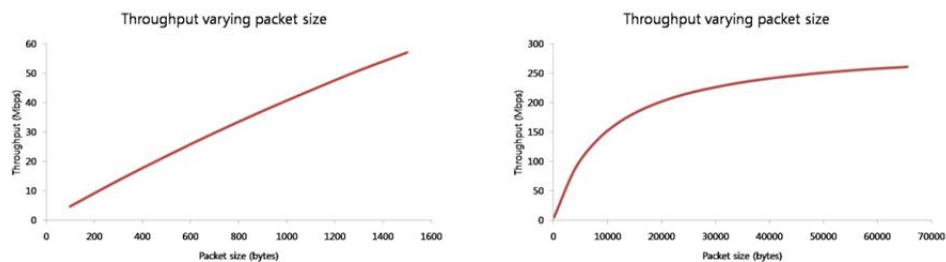


Figure 1 shows the throughput of IEEE 802.11n varying packet size. Although the channel bit rate is 300Mbps, the throughput is only approximately 57Mbps when the packet size is 1500 bytes (which is the maximum transfer unit for Ethernet). The degradation of throughput is caused from PHY and MAC overhead. Figure 2 explains the reason. To transmit a packet, a node needs to wait for DIFS after the channel becomes idle, wait for $67.5\mu\text{s}$ in average for backoff, before transmitting the packet. When a node transmits a packet, it first sends a preamble and a PLCP header, which takes $32\mu\text{s}$ in the Greenfield mode with 2×2 MIMO. Then, the 1500 bytes of data is sent, which takes $40\mu\text{s}$ with 300Mbps channel bit rate. Followed by SIFS of $10\mu\text{s}$, the receiver sends an ACK, which also requires preamble and PLCP header. The total time required for one packet transmission is $210.7\mu\text{s}$, which calculates to 57Mbps of data rate.

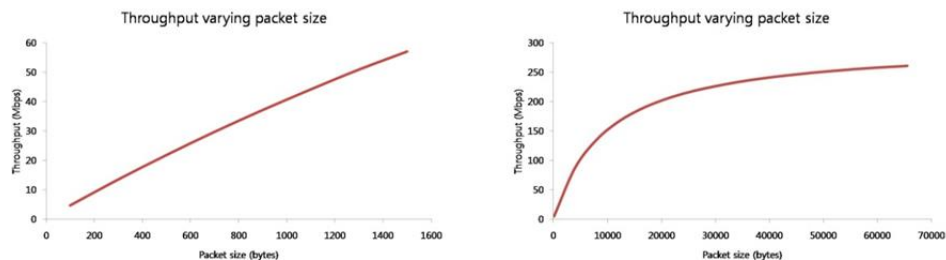


Figure 1 also shows that with packet aggregation, the throughput can increase to near the maximum. With 65,536 bytes of packet size, the effective throughput becomes 260Mbps.

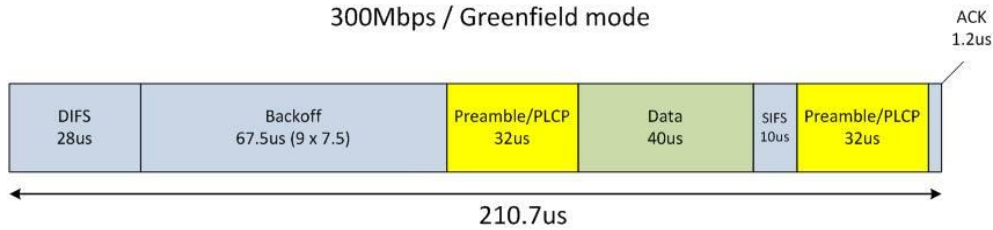


Figure 2. The time required for transmission of data in IEEE 802.11n Greenfield mode with 300Mbps channel bit rate

Now we consider dividing a single channel into multiple narrow channel. The 300Mbps of channel bit rate is achieved using a 40MHz channel. If we use a 20MHz channel, we get 144.4Mbps, using the same modulation and coding. The reason that a 20MHz channel does not achieve half the bit rate of a 40MHz channel is because of the guard band (as shown in Figure 3). Thus, there is a capacity loss when we divide a channel into multiple narrow channels. The timing parameters such as symbol duration also changes with channel width [8].

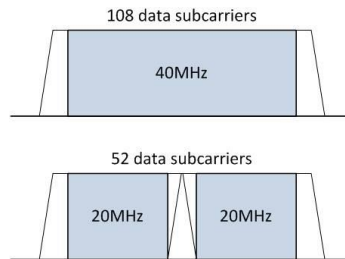


Figure 3. When dividing a single channel into multiple channels, a guard band is required to avoid interference, resulting in capacity loss

For example, when we divide a 40MHz channel into two 20MHz channel, we lose approximately 4% of the channel capacity. However, the gain from reducing MAC overhead is greater than the capacity loss.

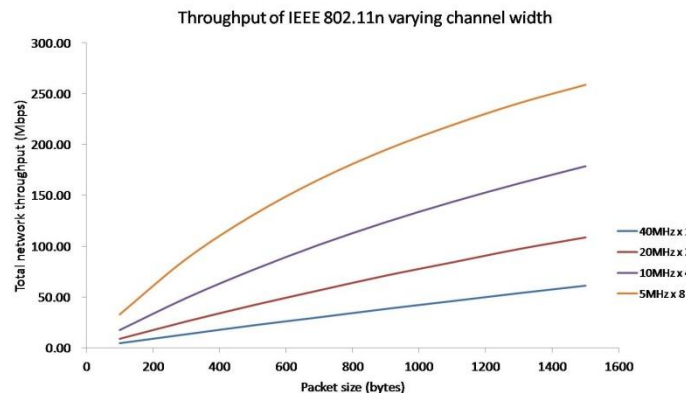


Figure 4. Throughput of IEEE 802.11n varying channel widths

Using simulation, we have plotted the throughput of IEEE 802.11n for different channel widths. The channel bit rate of 40MHz, 20MHz, 10MHz and 5MHz channels are 300Mbps, 144.4Mbps, 69.5Mbps, and 33.5Mbps, respectively. For 10MHz and 5MHz channels, we assumed the capacity loss from dividing channel is 4%. Other parameters are the same as the previous simulation. The result is shown in Figure 4. When the packet size is as small as 100 bytes, using eight 5MHz channels achieves 10 times the throughput compared to using a single 40MHz channel, even with the capacity loss. When the packet size is 1500 bytes, using eight 5MHz channels achieves approximately 4 times the throughput of a 40MHz channel.

When using eight 5MHz channels, all channels operate independently according to the IEEE 802.11 DCF protocol. This means that to transmit on each channel, a node needs to contend with other nodes using random backoff. However, when multiple channels are present, it may be possible to use the result of contention from one channel and remove the overhead of contention on other channels. This observation leads to our proposed technique, which is explained in detail in the next section.

3. Proposed Techniques

If nodes have multiple radios that can operate on different channels independently, there are MAC and routing protocols proposed to work with this kind of hardware capability [10, 11], especially for multi-hop networks. However, here we focus on using multiple radios (or a compound radio [7]) to reduce MAC overhead. The idea of the first technique is simple: If a node is occupying channel 1, it can also access channel 2 without contention assuming that other node will not access channel 2 if they are not occupying channel 1. In other words, the node occupying channel 1 also has the priority to access other channel. The status of channel 1 is used as the indicator for the access of other channels. Assuming the transmission range is the same for all narrow channels, the node accessing channel 2 will not experience collision because all neighbors will refrain from accessing the channel.

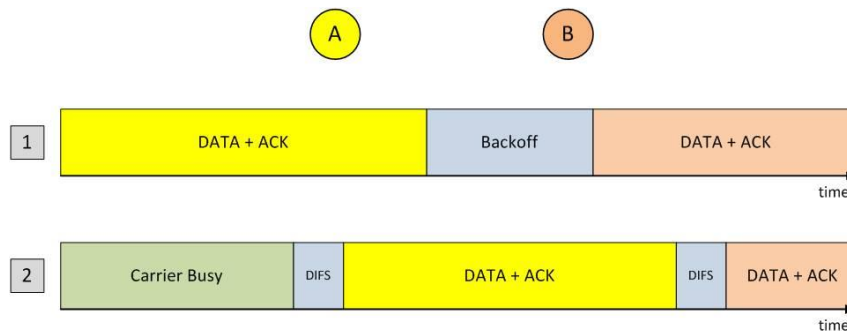


Figure 5. Behavior of nodes when the proposed technique is applied. When channel 2 becomes idle, each node checks the status of channel 1. Since node A is occupying channel 1, it accesses channel 2 without backing off. Node B behaves according to DCF

To implement this idea, one of the narrow channels is selected as the *contention channel*. The standard IEEE 802.11 DCF runs on this contention channel. On other channels, a node accesses the channel without contention if it is currently occupying the contention channel. The question is: how does a node decide whether it is occupying the contention channel or not? To answer this question, consider Figure 5. In the figure, there are two node and two channels. Channel 1 is the contention channel. Now suppose node A wins the contention on channel 1 and thus sends its data to its receiver. Then, it will receive an ACK from the receiver. Once node A receives the ACK, it goes into a random backoff period. Once backoff is done, meaning its backoff counter becomes zero, the node can transmit packet on the channel again. In Figure 5, after node A transmits packets on channel 1, node B acquires the channel. Now consider channel 2. Suppose channel 2 is busy for a while, but then becomes idle at some point. When the channel becomes idle, each node decides whether I can access channel 2 immediately, without contention. In Figure 5, at the point of time channel 2 becomes idle, it is node A that is occupying channel 1. Node 1 knows that it is currently using the channel, so it accesses channel 2 without random backoff. Node B, since it is not using channel 1, behaves according to the standard DCF protocol.

What if channel 1 is idle because all nodes are backing off? If we approach conservatively, no one should have priority on channel 2 and all nodes will contend for the channel as usual. However, we take an aggressive approach, and let the node who accessed the contention channel last has the priority on other channels, before any other node starts transmitting on the contention channel. Thus, in Figure 5, node A has the priority on other channels until node B starts transmitting on channel 1.

When the number of nodes increases, collisions start to occur on channel. If two nodes simultaneously transmit on channel 1, the packets collide. When the propose technique is applied, the problem is propagated to other channels, since multiple nodes will think they are occupying channel 1 and access other channels without contention. Thus, collision on the contention channel will mean collision on all other channels. To avoid this collision propagation problem, we propose the second technique. Consider Figure 6. In the figure, node A and node B transmits their packets at the same time, causing a collision. Other nodes in the vicinity of nodes A and B detects that there is a collision on channel 1. Suppose node C was sending a data to node D on channel 2. While receiving the packet on channel 2, node D detects a packet collision on channel 1. In this case, node D sets the *collision flag* in the ACK packet. When the ACK is sent, node A and B will receive the ACK. On seeing the collision flag set, the nodes do not immediately access channel 2 after it finishes receiving ACK. They follow the normal DCF procedure until it receives another ACK with the collision flag unset.

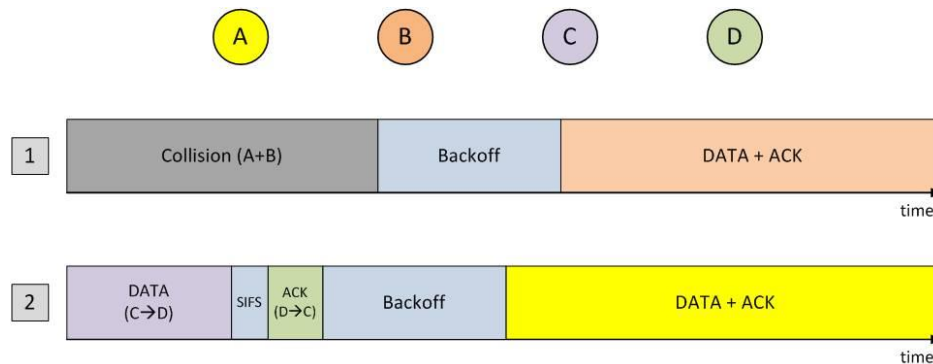


Figure 6. Behavior of the second technique when there is a collision on the contention channel. The receiver on channel 2 observes that there is a collision on channel 1, and sets the collision flag in its ACK so that nodes on channel 2 do not access the channel without contention

4. Performance Evaluation

We use the ns-2 simulator [12] to evaluate performance of the proposed techniques. We compare the three protocols. The first protocol is the normal IEEE 802.11 DCF. The second protocol is the first proposed technique which we call the *contention removal*, where a node occupying channel 1 accesses other channels without backing off. The third protocol uses the *collision avoidance* technique to reduce collisions on channels other than the contention channel.

In the simulation, nodes are placed close enough so that all nodes can reach each other in a single hop. Among the nodes, half of the nodes are assigned as packet sources, and the other half nodes are assigned as destinations. The parameters are the same as described in Section 2.

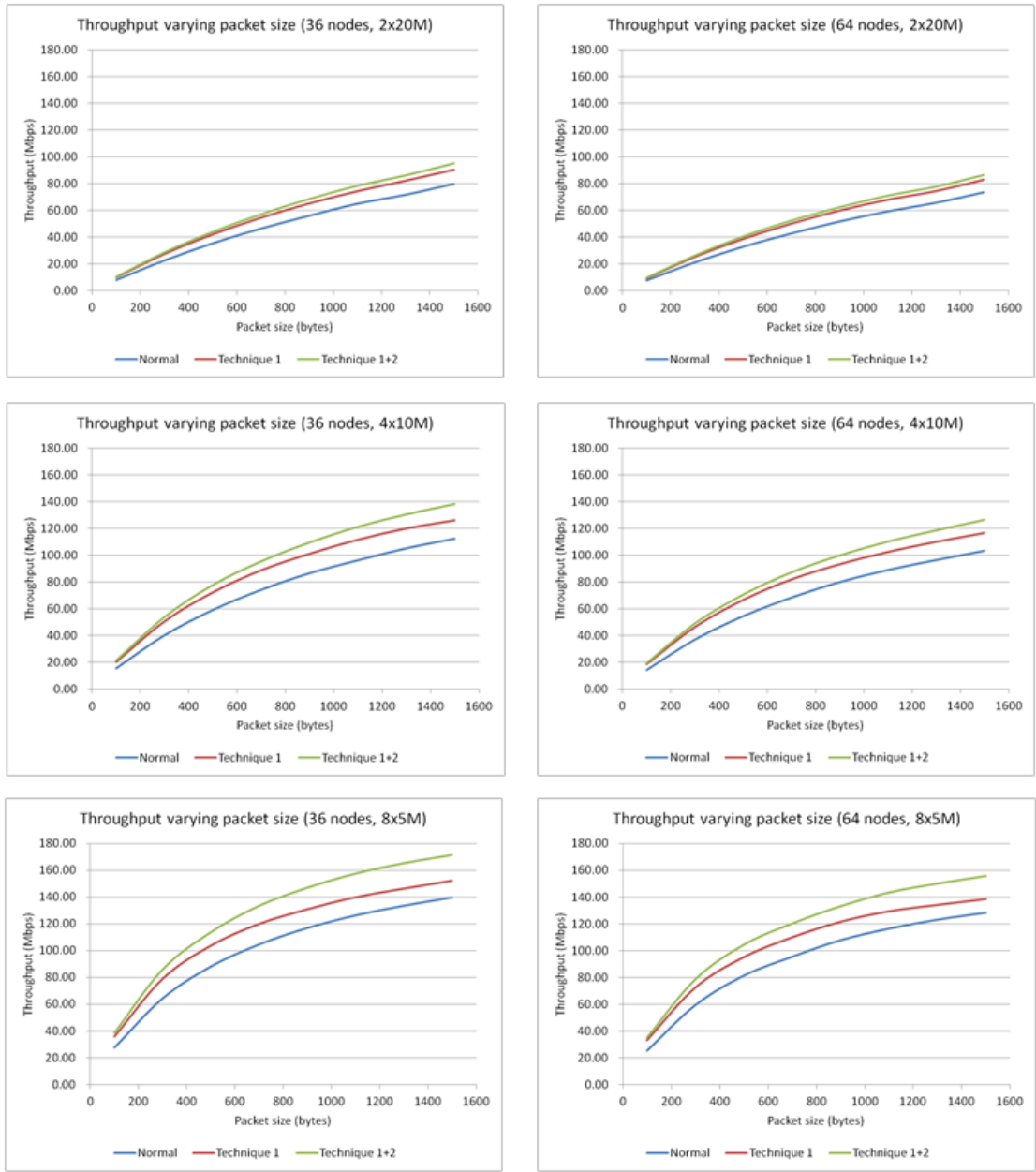


Figure 7. Network throughput varying packet size. The graphs differ in number of nodes and the configuration of channels

The result is shown in



Figure 7. The three graphs on the left are when the number of nodes is 36, and the three graphs on the right is when the number of nodes is 64. When using two 20MHz channels, the standard DCF achieves 79.9Mbps with 1500-byte packets. Applying the contention removal technique improves the throughput to 90.4Mbps (13% improvement over DCF), and applying the collision avoidance technique further improves the throughput to 95Mbps (18% improvement over DCF). When the packet size is 100 bytes, DCF achieves 8.3Mbps, the contention removal technique achieves 10.0Mbps (20%), and the collision avoidance technique achieves 10.4 (26%). With eight 5MHz channels and 1500-byte packets, contention removal and collision avoidance achieve 10% and 23% more throughput compared to DCF, respectively. When the packet size is 100 bytes, contention removal and collision avoidance achieve 30% and 40% more throughput compared to DCF, respectively. The proposed techniques improve throughput over DCF regardless of channel width or packet sizes. However, the effectiveness of the proposed techniques is maximized when the number of channels is large and the packet size is small. Overall, the proposed techniques improve the efficiency of IEEE 802.11 wireless networks by effectively reducing the MAC overhead.

5. Conclusion

MAC overhead is a major source of throughput degradation in high-rate IEEE 802.11-based wireless networks. Using multiple narrow channel instead of a single wideband channel reduces the MAC overhead and improves efficiency. In this paper, we further improve resource efficiency by removing contentions from on the narrow channels. Contention is done only on a selected narrow channel called contention channel, and other channels use the status of this contention channel to determine who can access the channel. To avoid propagation of collisions after applying this technique, we also propose a collision avoidance mechanism where a node detecting collision on the contention channel notifies the nodes on other channels, so that nodes do not collide on the other channel. Simulations reveal that the proposed techniques improve throughput by a considerable amount.

Acknowledgements

This work is supported by Hallym University Research Fund 2013 (HRF-201303-015).

References

- [1] IEEE 802.11 Working Group, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification", IEEE Std. 802.11-2012, (2012).
- [2] 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).
- [3] Y. Lin and V. Wong, "Frame Aggregation and Optimal Frame Size Adaptation for IEEE 802.11n WLANs", in Proc. of IEEE GLOBECOM, (2006).
- [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] E. Magistretti, K. Chintalapudi, B. Radunovic and R. Ramjee, "WiFi-Nano: Reclaiming WiFi Efficiency Through 800ns Slots", in Proc. of ACM MobiCom, (2011).
- [6] S. Sen, R. Choudhury and S. Nelakuditi, "No Time to Countdown: Migrating Backoff to the Frequency Domain", in Proc. of ACM MobiCom, (2011).
- [7] K. Chintalapudi, B. Radunovic, V. Balan and M. Buettner, "WiFi-NC: WiFi Over Narrow Channels", in Proc. of USENIX NSDI, (2012).
- [8] R. Chandra, R. Mahajan, T. Moscibroda, R. Raghavendra and P. Bahl, "A Case for Adapting Channel Width in Wireless Networks", in Proc. of ACM SIGCOMM, (2008).
- [9] X. Yang and N. Vaidya, "Spatial Backoff Contention Resolution for Wireless Networks", UIUC Technical Report, (2006).

- [10] P. Kyasanur and N. Vaidya, "Routing and Link-layer Protocols for Multi-Channel Multi-Interface Ad Hoc Wireless Networks", in Mobile Computing and Communications Review, vol. 10, no. 1, **(2006)**, pp. 31-43.
- [11] R. Draves, J. Padhye and B. Zill, "Routing in Multi-radio Multi-hop Wireless Mesh Networks", in Proc. of ACM MobiCom, **(2004)**.
- [12] The Network Simulator ns-2, <http://www.isi.edu/nsnam/ns>.

Authors



Jungmin So

He received the B.S. degree in computer engineering from Seoul National University in 2001, and Ph.D. degree in Computer Science from University of Illinois at Urbana-Champaign in 2006. He is currently an assistant professor in Department of Computer Engineering, Hallym University. His research interests include wireless networking and mobile computing.



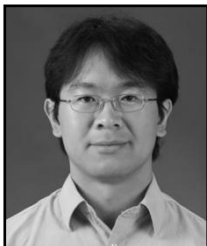
Cheal-Hwan Yoon

He graduated from Sokcho high school in 2006. He is currently an undergraduate student in Department of Computer Engineering, Hallym University. His research interests include wireless networking and mobile computing.



Joo-Yub Lee

He received the B.S. in Computer Engineering from Hallym University in 2013. He is currently a Graduate student in Department of Computer Engineering, Hallym University, pursuing a Master's degree. His research interests include wireless networking and large-scale systems.



Jeong-Gun Lee

He received the B.S. degree in computer engineering from Hallym University in 1996, and M.S. and Ph.D. degree from Gwangju Institute of Science and Technology (GIST), Korea, in 1998 and 2005. He is currently an assistant professor in the Computer Engineering department at Hallym university.



Young Woong Ko

He received both a M.S. and Ph.D. in computer science from Korea University, Seoul, Korea, in 1999 and 2003, respectively. He is now a professor in Department of Computer engineering, Hallym University,

Korea. His research interests include operating system, embedded system and multimedia system..

