Use Your Frequency Wisely: Explore Frequency Domain for Channel Contention and ACK

Xiaojun Feng, Jin Zhang, Qian Zhang and Bo Li
Department of Computer Science and Engineering,
Hong Kong University of Science and Technology, Hong Kong
{xfeng, jinzh, qianzh, bli}@cse.ust.hk

***Abstract*—The promise of high speed (over 1Gbps) wireless transmission rate at the physical layer can be significantly compromised with the current design of 802.11 DCF. There are three overheads in the 802.11 MAC that contribute to the performance degradation: DIFS, random backoff and ACK. Motivated by the recent progress in OFDM and self-interference cancellation technologies, in this paper, we propose a novel MAC design called REPICK (REversed contention and PIggy-backed ACK) to collectively address all the three overheads. The key idea in our proposal is to take advantage of OFDM subcarriers in the frequency domain to enhance the MAC efficiency. In REPICK, we propose a novel reverse contention algorithm, which enables and facilitates receivers to contend channel in the frequency domain (reversed contention). We also design an efficient mechanism which allows ACKs from receivers to be piggy-backed through subcarriers together with the contention information (piggy-backed ACK). We prove through rigorous analysis that the proposed scheme can substantially reduce the overheads associated with 802.11 DCF and a guaranteed throughput gain can be obtained. In addition, results from extensive simulations demonstrate that REPICK can improve the throughput by up to 170%.**

I. Introduction

The data transmission rate at the physical layer in wireless networks can be up to 600Mbps using the latest 802.11n protocol with advanced techniques such as MIMO (Multiple-Input Multiple-Output). Future standards, such as 802.11ac, are expected to provide even higher data rates (>1Gbps). These, however, cannot be materialized without adequate medium access control (MAC) protocol. The efficiency of 802.11 protocol can drop dramatically with the increase of the data transmission rate and the actual throughput can be much lower than the physical throughput [1]. It was reported in [2] that only an actual throughput of 60 Mbps can be achieved with the 300 Mbps data rate.

As shown in Fig. 1, in the 802.11 MAC, there are three timing factors that do not contribute to the actual data transmission time: 1) DCF inter-frame spacing (DIFS), which is time spacing with the lowest priority in 802.11 DCF; 2) time domain contention and backoff, during which each node keeps silent for several randomly chosen time slots before data transmission; and 3) short inter-frame spacing (SIFS) and ACK, which are used for a receiver to send a small acknowledgement packet to a sender for the most recently received frame. Normally, the duration of DIFS is about $28-34\mu s$ [2]. The average backoff time depends on the size of the contention window (CW). Suppose the minimum CW size is 16. The time spent on backoff will be at least $72\mu s$. For the ACK transmission, at least a $20\mu s$ preamble is required regardless of the length of the ACK [5]. These three factors add up to $120\mu s$. Evidently, suppose the data rate is 300 Mbps and the size of a data packet is 1500 bytes, the data transmission time would be only $40\mu s$. In another word, in this case, the MAC layer efficiency is a merely 25%. It is obvious that the overhead in the 802.11 DCF has a huge influence on the overall throughput.

There have been many works in the existing literature to improve the efficiency of 802.11 MAC. Most of them focused on reducing the average backoff time. Some works proposed to use an optimal contention window size, such as [6] [7] and [8]. In [9], a semi-random backoff scheme was proposed to set a dedicate backoff time for each node to avoid collision and to reduce the average backoff slots. And in [1], multiple nodes form a lossy contention group to reduce the number of contention entities in one contention domain to improve the average backoff time. Recently, some works propose to use physical layer technologies to improve the MAC layer efficiency [3] [4]. Within this category, some works, for example, FICA [2], T2F [10], Back2F [11], propose to conduct channel contention in the frequency domain. They treat the OFDM (Orthogonal Frequency Division Multiplexing) subcarriers used at the physical layer as integer numbers and use these subcarriers for channel contention to reduce the backoff time. Another example is SMACK [5], in which each client uses a dedicated subcarrier to acknowledge the broadcast packet from the Access Point(AP) concurrently in order to reduce the overhead of multiple ACKs. In summary, the existing works have primarily tried to reduce one of the three overheads in the 802.11 MAC. On the contrary, our work in this paper, attempts to address all the three issues together in an integrated manner.

Before describing our key ideas, we first present two observations. The first is that for each sender−receiver pair, either the sender or the receiver can be responsible for channel contention. Conventionally, channel contentions are performed by senders. There is, however, one unique advantage if receivers are also allowed to do channel contention. If a receiver has acquired channel access for its sender, other senders within the vicinity of the receiver may not transmit, thus will not interfere the receiver. In other words, the well known hidden

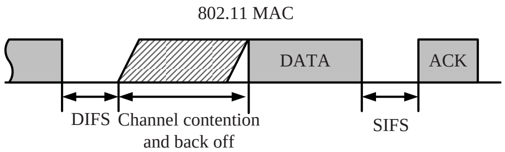


Fig. 1: Illustration of 802.11 MAC.

terminal problem can consequently be mitigated without the need of RTS/CTS. To further illustrate this, an example is given in Fig. 2, in which receiver R is contending for the link pair ($S \rightarrow R$) and the other two senders N_1, N_2 are contending for themselves. When R wins, N_1 and N_2 would have to keep silent. The second observation is that with the use of OFDM technology, more information can be encoded with OFDM subcarriers. As the standard evolves, wider bandwidth will provide more subcarriers. In 802.11g, only 48 data subcarriers is used. However in 802.11n, 104 data subcarriers are available in an aggregated 40MHz channel [13] [14]. In the future 802.11ac standard, there can be more than 200 subcarriers in an 80MHz channel. Moreover, with the technology of software defined radio, we can obtain 256 or more subcarriers within a 20 MHz channel [2]. As a result, we are able to encode both the contention and also the ACK information within one OFDM symbol.

Based on these observations, we propose a novel MAC design called REPICK (REversed contention and PIggy-backed ACK) based on two innovations. First, we allow a receiver to contend for its sender in the frequency domain (reversed contention). If a sender has more than one packets to send to the same receiver, after receiving the first packet successfully, the receiver will activate one randomly chosen subcarrier to contend channel for its sender's following packets in the frequency domain. Second, when contending, the receiver uses another subcarrier as an ACK for the latest received frame (piggy-backed ACK). In REPICK, the channel contention is done in the frequency domain, so the overhead of time domain contention and backoff can be mitigated and only the duration of an OFDM symbol ($< 10\mu s$) is required. With reversed contention, ACKs can be piggy-backed during channel contention, thus the overhead of ACK can be completely eliminated. Moreover, since ACK is integrated into contention, there is no need for different priorities of inter frame spacings. As a result, DIFS can be replaced by SIFS. In this way, all the three overheads in 802.11 DCF can be reduced.

The contributions of this paper are summarized below:

- We proposed a novel MAC protocol named REPICK. By utilizing information on subcarriers, REPICK uses reversed contention and piggy-backed ACK to reduce the overheads in traditional 802.11 MAC protocol.
- We carry out capacity analysis of REPICK, and prove its guaranteed throughput gain over 802.11 DCF.
- Through extensive simulations, we demonstrate REPICK's throughput gain over both 802.11 DCF and the state-of-the-art MAC protocol based on the idea of frequency domain contention.

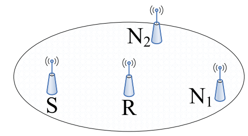


Fig. 2: Motivating example of REPICK.

The rest of the paper is organized as follows. In Sec. II, we introduce the design of REPICK, followed by the mathematical analysis of REPICK in Sec. III. In Sec. IV, we discuss and evaluate the performance of REPICK using simulation. Related works are discussed in Sec. V. Sec. VI concludes the paper.

II. Protocol Design

In this section, we first introduce the basic ideas of REPICK and then describe its major components.

A. REPICK Basic Design

We assume OFDM is used at the physical layer. The wireless channel is divided into multiple subcarriers. Data can be transmitted through subcarriers in parallel. We also assume that each node has two antennas: transmission antenna and listening antenna. With self-interference cancellation technologies, the listening antenna can detect which subcarriers are activated by nearby nodes when the transmission antenna is sending packets concurrently. The feasibility of the use of the listening antenna has been verified by system implementations in existing works such as [10] [11] [12].

The basic idea of REPICK is: *to leverage information on subcarriers to conduct both channel contention and ACK in the frequency domain concurrently*.

There are several challenges to overcome. First, how can we distinguish ACKs from different receivers? There may be multiple sender-receiver pairs within one contention domain. We say nodes are in the same contention domain if they can mutually overhear/interfere with each other. Second, how can a sender knows whether its receiver has won the channel? The sender should also know which contending subcarrier is chosen by its receiver. Third, how can REPICK survive from the hidden node problem? When there are multiple contention domains, hidden nodes are very likely to exist.

REPICK has three major components to tackle the three challenges respectively.

1) An algorithm to identify each node with a uniquely assigned identification subcarrier. In REPICK, subcarriers are partitioned into two parts: Identification Subcarriers and Contention Subcarriers. Each node is assigned with a distinct subcarrier from the identification subcarriers. In any contention domain, there should not be two nodes sharing the same identification subcarrier. In this paper, we just focus on the scenario of a sparse to medium network with no more than 16 nodes in one contention domain. We also assume that nodes are static and will not dynamically join or leave the network. We will address the dense network scenario and the case of dynamically joining/leaving in our future work.

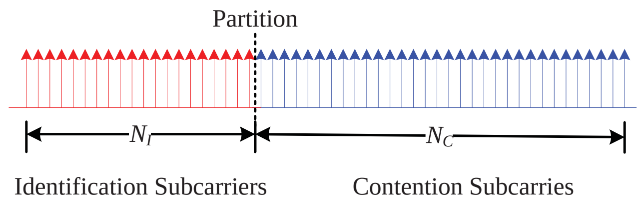


Fig. 3: Partition of Subcarriers.

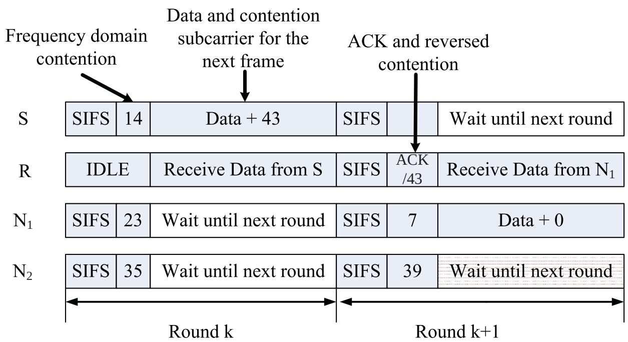


Fig. 4: Illustration of REPICK access method.

2) A round-based schedule for reversed contention and piggy-backed ACK. In REPICK, the procedure of data transmission is divided into rounds. For the first packet, the contention is done by the sender. If the sender has more data to transmit, it will inform its receiver about the next contention subcarrier to use with an additional byte in the data packet. After the packet transmitted, the receiver uses this subcarrier for channel contention and concurrently activates the sender's identification subcarrier as an ACK.

3) A round-based transmission retreat scheme. In REPICK, if hidden nodes are detected by the sense of missing ACKs, the sender will keep silent for one or more rounds instead of backoff for multiple time slots.

B. Subcarrier Partitioning

We assume there are totally N_S subcarriers in one wireless channel. We partition all the subcarriers into two parts (Fig. 3). The first N_i subcarriers are used for node identification and ACK. According to our assumption, $N_i \leq 16$. And the other $N_C = N_S - N_i$ subcarriers are used for frequency domain contention.

We use a multi-coloring scheme for ID subcarrier distribution. We first construct a un-directional graph $G(V, E)$ to represent the neighboring relationship among nodes. In $G(V, E)$, V denotes all nodes in the network. If two nodes i and j are within the communication range of each other, there is an edge $e(i, j) \in E$. Base on G, we further construct a graph $G'(V', E')$ to represent nodes' ID conflict relationship. For each vertex i in V, there is an i' in V', for each edge pair $e(i, j)$ and $e(j, k)$ in E, there are three edges $e'(i', j')$, $e'(j', k')$ and $e'(i', k')$ in E'. The reason why there is an $e'(i', k') \in E'$ is because node i and node k are both neighbors of j and they are both in the contention domain of j. Finally, we use a total of N_i colors to do vertex coloring in G'. Each color is corresponding to one ID subcarrier. If we can find a valid vertex coloring in G', we then assign each node an identical ID subcarrier according to the coloring result.

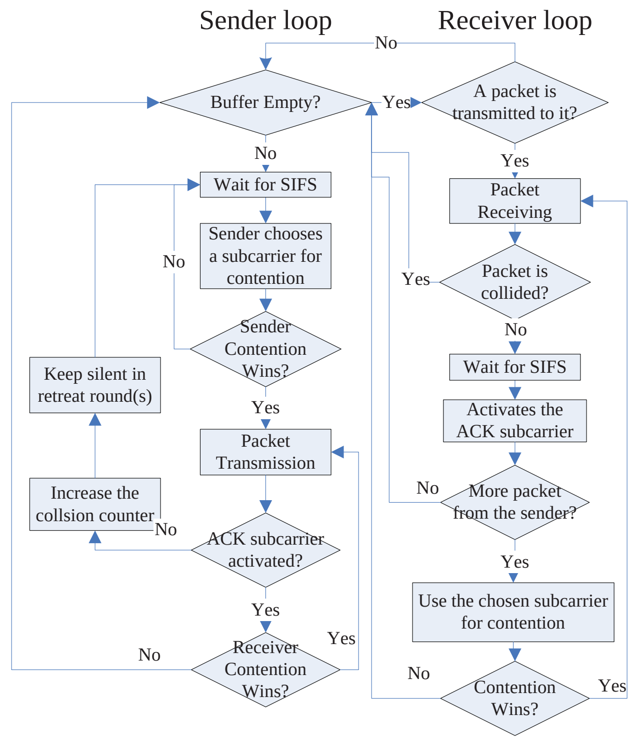


Fig. 5: Flow chart of the MAC protocol of REPICK.

C. Reversed Contention and Piggy-backed ACK

Data transmission in REPICK is divided into rounds. In each round, there are three parts: SIFS, Contention/ACK and Data transmission. Take the topology in Fig. 2 for example, we assume three nodes S, N_1 and N_2 are trying to transmit packets to node R. All four nodes are in the same contention domain. The protocol behavior can be illustrated in Fig. 4.

At the beginning of each round, all senders wait for an SIFS. Then each sender activates a randomly chosen contention subcarrier with ID in the range $[1, N_C]$. In Fig. 4, S, N_1 and N_2 choose contention subcarrier 14, 23 and 35 respectively. The transmission on the contention subcarrier can be implicitly synchronized and detected by the listening antenna [10] [11]. The node with the smallest contention subcarrier ID wins. Here, S wins and starts data transmission to R. Meanwhile, other nodes keep silent until the end of this round. In the data packet from S to R, there is an additional byte indicating the contention subcarrier for the next round. If the byte is 0, it means that a sender has no more packets to send. In our example, S chooses 43. After R receiving the packet, it activates S's identification subcarrier for ACK. Concurrently, R uses 43 for channel contention. In our example, S knows that R loses by checking that 43 is not the smallest contention subcarrier and it will keeps silent. Meanwhile, N_1 wins and it start transmission in round $k + 1$.

A complete flow chart of the REPICK is shown in Fig. 5.

D. Transmission Retreat

Channel contention in REPICK is conducted in the frequency domain, there is no time domain backoff needed in REPICK. However, when there are multiple contention domains, hidden nodes may exist. Suppose two nodes N_1, N_2 are in different contention domains. When they both try to send packets to R. In the view point of N_1 and N_2, there are no other nodes contending with them. Therefore, in every round, they win the channel and transmit data to R simultaneously. However, R cannot decode any either data packet because of collision. In this case, one of the senders needs to hold its transmission to avoid collision. This procedure is different from the time domain backoff in 802.11 DCF. Since a sender holds its transmission for one or more rounds instead of several time slots, we call it *Transmission Retreat*.

In REPICK, every sender maintains a counter to record the number of collisions detected. When there is no ACK from its receiver, a sender finds out that a collision happened. Then, it increases the counter and chooses a random number from $[0, counter]$ to be the number of rounds for transmission retreat. There is a maximum value of the counter, say K_{max}. K_{max} is just like the maximum contention window size in 802.11 DCF. As soon as a data packet is successfully transmitted, the counter will be reset to zero. Since K_{max} plays an important role in the transmission retreat scheme, we should choose its wisely. Too large K_{max} may cause a waste of the wireless media while too small K_{max} may lead to additional collisions. Through tests, we believe $K_{max} = 3$ is good enough to achieve a balance between channel idle and collision. The choice of optimal K_{max} will be left as our future work.

III. Performance Analysis

In this section, we use mathematical analysis to evaluate the performance of REPICK. We first calculate the collision probability of frequency domain contention. Then we analyze the property of REPICK in both single and multiple contention domains.

A. Collision Probability of Frequency Domain Contention

Collision happens when two or more nodes consider themselves winners in channel contention. In other words, these nodes all choose the same smallest contention subcarriers among all nodes. Suppose the number of nodes in a contention domain is N_0. We assume the probability of collision, when the index of the smallest subcarrier is i, is $P(i)$. Such that we have:

$$P(i) = \sum_{j=2}^{N_0} \binom{N_0}{j} \left(\frac{1}{N_C}\right)^j \left(\frac{N_C - i}{N_C}\right)^{N_C - j}$$

So the total collision probability P_C can be calculated as:

$$\begin{aligned} P_C &= \sum_{i=1}^{N_C} [P(i)] \\ &= \sum_{i=1}^{N_C} \left[\sum_{j=2}^{n} \binom{n}{j} \left(\frac{1}{N_C}\right)^j \left(\frac{N_C - i}{N_C}\right)^{N_C - j} \right] \end{aligned} \quad (1)$$

B. System Capacity in Single Contention Domain

In single contention domain, at most one sender can successfully send out data packet. Suppose the size of each data packet is P_{DATA}, the time of SIFS is T_{SIFS}, the time for frequency domain contention is T_{CONT} and the data transmission rate is R_{DATA}. We can calculate the time for data transmission as:

$$T_{DATA} = P_{DATA}/R_{DATA}$$

Such that the capacity of REPICK in one collision free round can be expressed as:

$$C_{NO\_COLL} = \frac{P_{DATA}}{T_{SIFS} + T_{CONT} + T_{DATA}}$$

Since the probability of one collision free round is $1 - P_C$, thus the system capacity of REPICK in single contention domain can be express as:

$$C_{REPICK} = (1 - P_C) \cdot \frac{P_{DATA}}{T_{SIFS} + T_{CONT} + T_{DATA}} \quad (2)$$

To ensure all nodes in the same contention domain can overhear the signal, the duration of T_{CONT} should be at least one FFT window [10]. Here we set:

$$T_{CONT} = T_{FFT} + 2T_{prop}$$

T_{FFT} is the time frame of an FFT window, and T_{prop} is the time for the radio signal to reach the maximal distance of the network. T_{FFT} is usually less than $10\mu s$ for $20MHz$ 802.11 channel.

For comparison, we use the result from [15] as an estimation of the capacity of 802.11 MAC:

$$\begin{aligned} C_{802.11} &= \frac{2(1-p)}{2-p} \\ &\times \frac{P_{DATA}}{T_{DIFS} + \frac{CW_{min}}{N_0+1} + T_{DATA} + T_{SIFS} + T_{ACK}} \end{aligned} \quad (3)$$

In Equ. 3, CW_{min} is the minimum contention window size and p is probability of a collision for 802.11. According to [15]. p can be expressed as:

$$p = 1 - \left(1 - \frac{2(1-2p)}{1 - p - p(2p)^m} \frac{1}{W}\right)^{n-1} \quad (4)$$

where the meaning of m is that $CW_{max} = CW_{min} \times 2^m$. When $CW_{min} = 16$ and $CW_{max} = 1024$, $m = 6$.

Therefore, the throughput gain of REPICK over 802.11 MAC can be expressed as:

$$G_S = \frac{C_{REPICK}}{C_{802.11}} - 1 \quad (5)$$

Claim 1. *In single contention domain, the capacity of REPICK outperforms 802.11 for all data rates higher or equal than 6Mbps, given the typical parameters of 802.11 in table I and* $N_C \geq 48$.

Proof: With Equ. 2,3, and 5, G_S can be rewritten as:

$$G_S = \frac{(1 - Pc)/(O_R + T_{DATA})}{2(1-p)/(2-p)(O_8 + T_{DATA})} - 1$$

TABLE I: Timing parameters of 802.11.

T_{SLOT}	$9\mu s$
T_{DIFS}	$28\mu s$
T_{SIFS}	$10\mu s$
T_{FFT}	$10\mu s$
T_{prop}	$1\mu s$
T_{ACK}	$26\mu s$
P_{DATA}	1500 Bytes
N_0	≤ 16
CW_{min}	16
CW_{max}	1024

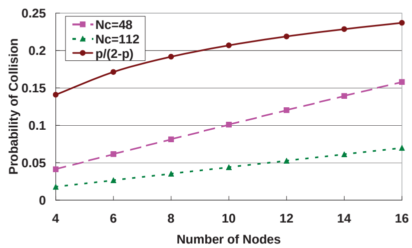


Fig. 6: Required collision probability in REPICK.

in which O_8 and O_R are overheads in 802.11 and REPICK respectively. With the typical parameters in Table I, we have:

$$O_8 = T_{DIFS} + \frac{N_0}{CW_{min}+1} + T_{SIFS} + T_{ACK} \approx 66\mu s$$

And the overhead in REPICK

$$O_R = T_{CONT} + T_{SIFS} \approx 22\mu s$$

Since G_S decreases as T_{DATA} increases, to guarantee that REPICK outperforms 802.11 in all data rates, we just need $G_S > 0$ for the lowest data rates. suppose the packet size is 1500 Bytes and the lowest data rate supported is 6Mbps. Then $T_{DATA} \leq 1500 \times 8/6M = 2000\mu s$. Now we need,

$$\frac{(1-P_C)(2-P)(O_8+2000)}{2(1-p)(O_R+2000)} - 1 > 0$$

Since both O_8 and O_R can be ignored compared with $2000\mu s$, we have

$$P_C <= 1 - \frac{2(1-p)}{2-p} = \frac{p}{2-p} \tag{6}$$

To calculate p from Equ. 4, we need to conduct numerical test. In Fig. 6, we plot the required collision probability and P_C in various N_C. It's easy to see that, when $N_C \geq 48$, the P_C is less than $p(2-p)$ which guarantees that REPICK has higher capacity than 802.11. ■

Although Claim 1 is based on a particular parameter set, it provides an overview of the capacity of REPICK in single contention domain. Equ. 6 can also be used as a rule for the selection of N_C and N_i to guarantee the throughput gain over 802.11.

C. Hidden Nodes in Multiple Contention Domains

In the case of multiple contention domains, hidden node problem may exist. Hidden nodes are nodes that within the interference range of a receiver, but outside the carrier sense

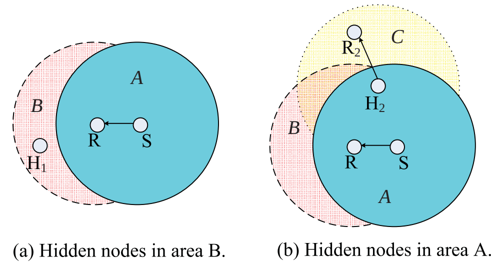


Fig. 7: Different cases of hidden nodes.

range of the sender. When there are hidden nodes, transmission may fail because of collision. Hidden nodes are very common for 802.11 DCF without RTS/CTS. Let us consider the scenario in Fig. 7a. The blue area A is the contention domain of sender S and the red area B is in receiver R's contention domain, but out of S's contention domain. When S sends data to R, nodes in area B cannot sense S's transmission. Therefore, as long as there is another node such as H_1 in B willing to send data during the data transmission period of link $S \to R$, a collision will happen. Suppose nodes are uniformly distributed and there are N_0 nodes in every contention domain and the radius of a contention domain is R_0. We assume every station has packets to send. With the method proposed in [16], the number of nodes in area B can be estimated as:

$$N' = 1.3\frac{N_0}{\pi R_0^2} \cdot R_0^2 \approx 0.41N_0$$

So the number of hidden nodes of S at any time S wants to transmit can be calculated as:

$$N^H_{802.11} = N' \cdot p = 0.41N_0 p \tag{7}$$

in which p is the collision probability of 802.11.

In REPICK, data transmission is divided into rounds, and contention can be performed by either a sender or a receiver. In the remaining of this section, We first analyze the probability that a contention is performed by a sender and a receiver respectively. Then we analyze the expected number of hidden nodes in REPICK.

1) The Probability of Sender Contention and Receiver Contention: In REPICK, to transmit the first packet between any link, a sender performs contention. If it wins and transmits the first packet successfully, its receiver will perform contention for follow-up packets. After the receiver contention, the sender transmits packet if it thinks its receiver wins. We can model the behavior of contention as a first order Markov process. And the state transition graph is illustrated in Fig. 8.

In Fig. 8, P_{win} is the probability of a node wins in the sender's contention domain. We have:

$$P_{win} = \sum_{i=1}^{N_C} \frac{1}{N_C} \left(\frac{N_C - i + 1}{N_C}\right)^{N_0 - 1} \tag{8}$$

.

Suppose in stable state, the probability of sender (receiver) contention is P_{sender} ($P_{receiver}$). We have:

$$\begin{aligned} P_{sender} + P_{receiver} &= 1 \\ P_{sender} \cdot P_{win} + P_{receiver} \cdot P_{win} &= P_{receiver} \end{aligned}$$

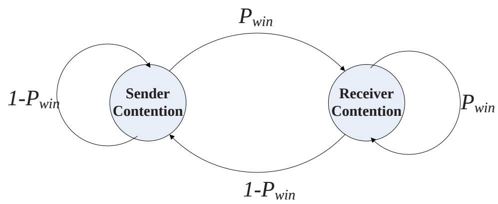


Fig. 8: State transition between sender contention and receiver contention.

By solving the above equations set, we have:

$$P_{sender} = 1 - P_{win} \tag{9}$$

$$P_{receiver} = P_{win} \tag{10}$$

2) Hidden Nodes When Sender Contention: After sender S wins, there are two possible kinds of hidden nodes. The first case is another sender lying in area B in Fig. 7a, such as H_1. Since H_1 cannot sense S, if H_1 also wins in its contention domain, it becomes a hidden node. The second case is another sender lying in area A and within the interference range of R, such as H_2 in Fig. 7b. In the second case, the receiver R_2 of H_2 lies in area C which is outside of the contention domain of S. R_2 does receiver contention and wins in the contention for H_2 and it also defeats S (but S doesn't know). Therefore, in the H_2's point of view, R_2 wins, such that H_2 starts data transmission and it becomes a hidden node for S.

The second case of hidden node depends on the location of a receiver and its sender, but is bounded by the number of receiver contention in area A.

So the expected number of hidden nodes when sender performs contention: $N^H_{REPICK\_sender}$ is bounded by:

$$\begin{aligned} & N^H_{REPICK\_sender} \\ \leq\ & N' \cdot P_{sender} \cdot P_{win} + (N_0 - N') \cdot P_{receiver} \cdot P_{win} \\ =\ & (0.41N_0 + 0.18N_0 P_{win}) \cdot P_{win} \end{aligned}$$

3) Hidden Nodes When Receiver Contention: Sender S considers its receiver R wins in the contention if the receiver chooses the smallest contention subcarrier in the sender's contention domain. But hidden nodes can also exist in the contention domain of R(e.g. H_2 in Fig. 7b). Similar with the case when sender contention, assuming the expected number of hidden nodes when receiver performs contention is $N^H_{REPICK\_receiver}$, it follows that:

$$N^H_{REPICK\_receiver} \leq N_0 \cdot P_{win} \cdot P_{win}$$

4) Hidden Nodes in REPICK: Consider both $N^H_{REPICK\_sender}$ and $N^H_{REPICK\_receiver}$, we can finally derive the expected number of hidden nodes for a link pair in REPICK.

$$\begin{aligned} & N^H_{REPICK} \\ =\ & N^H_{REPICK\_sender} \cdot P_{sender} + N^H_{REPICK\_receiver} \cdot P_{receiver} \\ =\ & 0.41N_0 P_{win} - 0.23N_0 P^2_{win} + 0.82N_0 P^3_{win} \end{aligned} \tag{11}$$

Now we have the following claim:

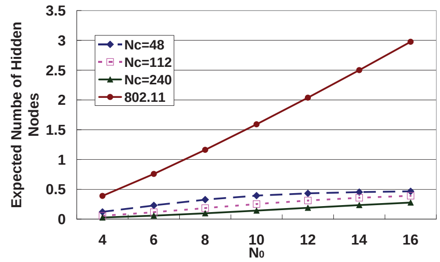


Fig. 9: Expected number of hidden nodes in 802.11 and REPICK.

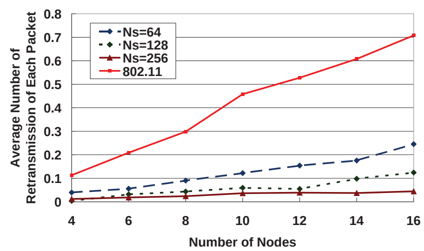


Fig. 10: Average number of retransmissions of each packet in single contention domain.

Claim 2. *Suppose nodes are uniformly distributed and there are multiple contention domains, the number of hidden nodes around a transmission pair $S \to R$ when applying REPICK is less then that when using 802.11 DCF, if $N_0 <= 16$ and $N_C \geq 48$.*

Proof: From Equ. 8, it's easy to see that P_{win} decreases as N_C decreases. Moreover, we have

$$\lim_{N_C \to \infty} P_{win} = \lim_{N_C \to \infty} \sum_{i=1}^{N_C} \frac{1}{N_C} \left(\frac{N_C - i + 1}{N_C} \right)^{N_0 - 1} = 0$$

It follows that:

$$\lim_{N_C \to \infty} N^H_{REPICK} = 0.$$

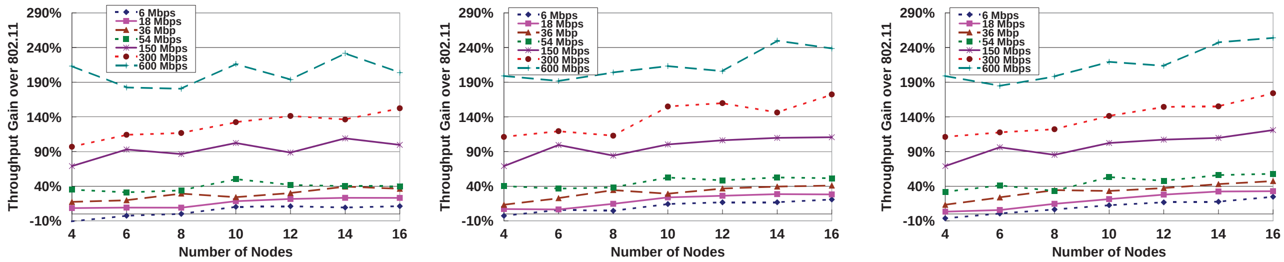
Therefore, when N_C is large enough, N^H_{REPICK} will be less than $N^H_{802.11}$. Again, through numerical testing, we can obtain the collision probability of 802.11 MAC p and also P_{win}. In Fig. 9, we plot $N^H_{802.11}$ and N^H_{REPICK} in various values of N_C. It's easy to see that, when $N_C \geq 48$, N^H_{REPICK} is less than $N^H_{802.11}$ for all $N_0 <= 16$. ■

IV. Simulation Evaluation

In this section, we use network simulations to evaluate the performance of REPICK in both single and multiple contention domains.

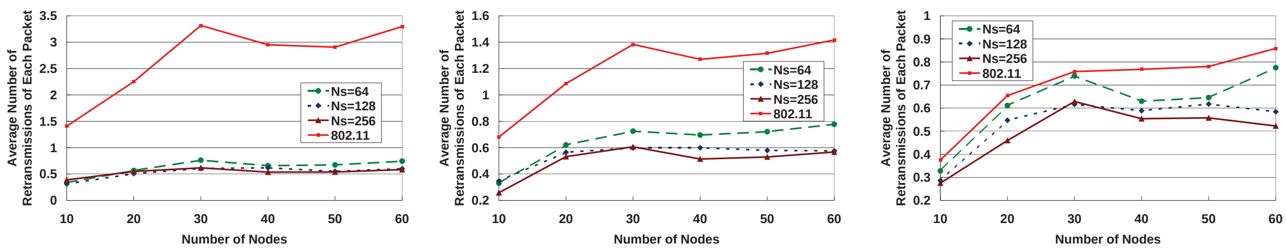
A. Performance of REPICK in Single Contention Domain

We first compare the performance of REPICK with 802.11 DCF in single contention domain with N_0 nodes. Throughout this section we mean "802.11" or "802.11 DCF" by the basic 802.11 CSMA/CA MAC protocol without CTS/RTS. N_0 ranges from 4 to 16. Nodes are distributed in a 50×50 area and



(a) $N_S = 64$. (b) $N_S = 128$. (c) $N_S = 256$.

Fig. 11: Throughput gain of REPICK compared with 802.11 in single contention domain.



(a) Data rate = 6Mbps. (b) Data rate = 54Mbps. (c) Data rate = 300Mbps.

Fig. 12: Average per-packet retransmissions of REPICK and 802.11 in different node densities.

each node can sense and decode packets from each other. We use the parameters in Table I for 802.11 DCF and set N_i to be 16 in REPICK. The data rate ranges from 6Mbps to 600Mbps. Each node randomly chooses a neighbor as receiver. We set each node to have the same number of packets to transmit. We repeat the simulation for 6 times for each node density and calculate both the average throughput gain and the number of retransmissions for different densities.

Fig. 10 shows the average number of retransmissions of 802.11 DCF and REPICK with different N_S. With larger N_S, N_C is also larger. Therefore, the number of collisions of REPICK is smaller. In all node densities, REPICK induces less retransmissions than 802.11 DCF.

Fig. 11 shows the system throughput of REPICK and 802.11 DCF in different N_S. In all cases, we observe that: 1), REPICK leads to throughput gain over 802.11 DCF in almost all cases (except for the case when data rate is 6Mbps and the number of nodes is less than 8); 2), the throughput gain over 802.11 DCF increases as the number of subcarriers increases, and 3) the throughput gain of REPICK over 802.11 DCF increases as the data rate increases.

These two figures verify our Claim 1 in Sec. III. REPICK achieves higher throughput than 802.11 DCF even with conventional number of subcarriers in 802.11g ($N_S = 64$). There are two reasons for the throughput gain of REPICK. First, in higher data rate, the ratio of overhead in 802.11 DCF is larger. Therefore, higher data rate leads to higher gain. Second, in REPICK, the probability of data collision is smaller. With larger number of contention subcarriers, higher gain is achieved.

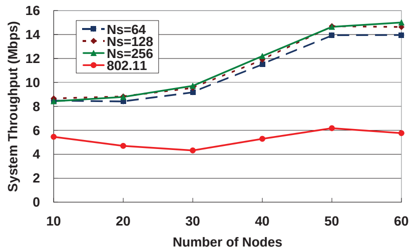
One thing to clarify is that, although REPICK has lower collision probability than 802.11 DCF, the throughput of REPICK may still be lower than 802.11 DCF especially in very sparse network and low data rate. The reason is that we set the number of packets to transmit as a finite number. After most of the nodes finished transmission, one collision leads to time domain transmission retreat for one or more rounds and degrades throughput of REPICK in these cases.

B. Performance of REPICK in Multiple Contention Domains

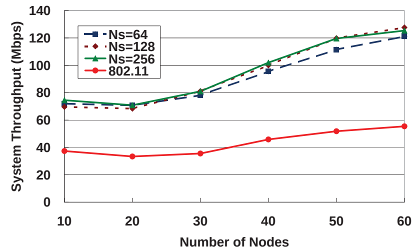
In the case of multiple contention domains, the number of nodes is set in the range [10, 60]. We set the communication range of each node to be 15, such that nodes may in different contention domains. For each node density, we test 12 different topologies. We calculate the average system throughput and the average number of retransmissions before a packet is successfully transmitted.

Here, we compare the performance of three MAC protocols: 802.11 DCF, REPICK and T2F [10]. T2F is a state-of-the-art MAC protocol exploiting the technique of frequency domain contention. The parameters for 802.11 DCF are the same as in Table I and $N_i = 16$ in REPICK. In T2F, we set the time of PIFS to be $19\mu s$ and a node cancels current data transmission and goes back to the beginning of each round if it waits too long for its predecessor to finish transmission. This is to ensure that the node is not starve when hidden nodes exist.

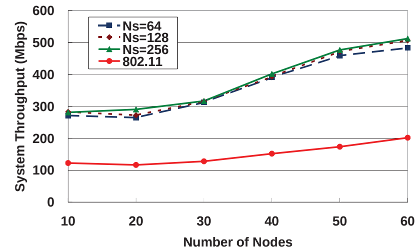
1) Number of Retransmissions of REPICK and 802.11 DCF: Fig. 12 shows the average number of retransmissions. In all the three data rates, REPICK has less retransmissions than 802.11 DCF, which means the probability of media access collision in REPICK is smaller. This result verify our Claim



(a) Data rate = 6Mbps.

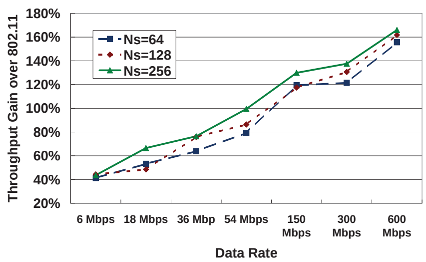


(b) Data rate = 54Mbps.

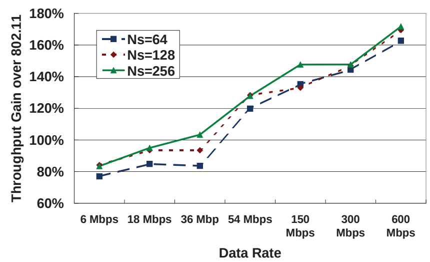


(c) Data rate = 300Mbps.

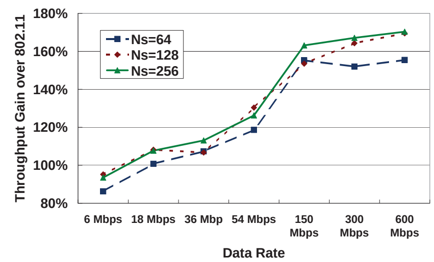
Fig. 13: System throughput of 802.11 DCF and REPICK in different node densities.



(a) 10 nodes.



(b) 30 nodes.



(c) 60 nodes.

Fig. 14: System throughput gain of REPICK over 802.11 DCF in different physical data rates.

2 in Sec. III. Also, we can see that with larger number of contention subcarriers, the number of retransmissions in REPICK is smaller. For both REPICK and 802.11 DCF, the retransmission number increases as the number of nodes increases because of hidden nodes. Another observation from Fig. 12 is that in higher data rates, the collision probability of 802.11 DCF is smaller while the collision probability of REPICK remains stable. That's because in 802.11 DCF, higher data rate means shorter data transmission time. Therefore, the probability of collision is lower. While in REPICK, the collision probability is determined mainly by the number of contention subcarriers.

2) Throughput of REPICK and 802.11 DCF: Fig. 13 shows the system throughput of REPICK and 802.11 DCF at three physical data rates, 6Mbps, 54Mbps and 300Mbps. In all cases, the system throughput of REPICK is significantly improved from 802.11 DCF due to less retransmissions. When node density increases, there maybe more contention domains and cause higher collision probability. So the throughput gain of REPICK over 802.11 DCF increases as the node density increases. When the number of nodes is larger or equal than 30, the throughput gains remain stable around 140%. since the number of contention domains in the network remains stable after the node density is high enough.

Fig. 14 shows the throughput gain of REPICK over 802.11 DCF. In all cases, the average throughput gain from REPICK is higher than 80%. With higher physical data rate, REPICK achieves higher throughput gain over 802.11 DCF, which verifies the benefit of REPICK in higher data rates.

3) Comparison between REPICK and T2F: T2F is a state-of-the-art design utilizing frequency domain contention. In each round, T2F determines a transmission order of three nodes. The nodes are then scheduled to transmit one after another after overhearing control messages. In T2F, node defers its transmission when it senses the channel busy. However, during the deferred period of transmission, other nodes may consider the scheduled transmission all finished and start to do channel contention again. This effect leads to the failure of the T2F schedule in multiple contention domains [11].

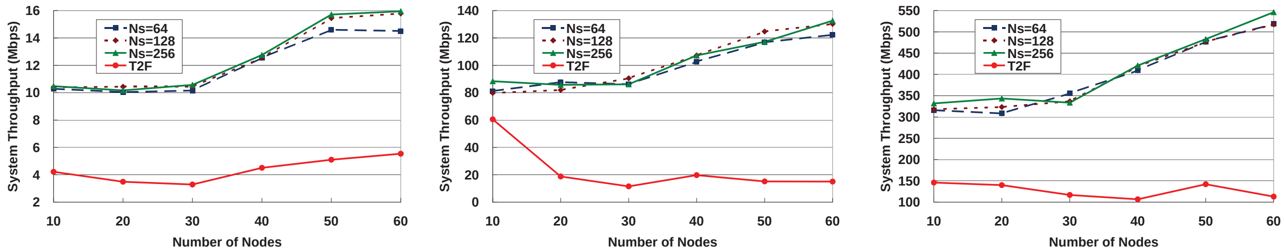
Fig. 15 shows the system throughput of REPICK and T2F with different node densities at 6Mbps, 54Mbps and 300Mbps data rate, respectively. The throughput gain of REPICK over T2F increases as the node density increases in all cases. When there are multiple contention domains, T2F is more likely to fail.

V. Related Works

There is a rich body of existing works in the literature trying to improve the efficiency of 802.11 DCF by reducing the overhead in random backoff [6] [7] [8] [1]. These works are mainly based on the time domain contention schemes. In contrast, REPICK further reduces the overhead of random backoff by utilizing diversity in the frequency domain.

There are also existing works taking advantage of the OFDM subcarriers in the frequency domain to improve MAC efficiency.

In [2], Tan et al. proposed to divide the entire 20MHz WiFi channel into multiple sub-channels to enable fine grained channel access to improve the MAC layer efficiency. Channel contention among clients are carried out in the frequency



(a) Data rate = 6Mbps. (b) Data rate = 54Mbps. (c) Data rate = 300Mbps.

Fig. 15: System throughput of REPICK and T2F in different node densities.

domain. However, their work requires a complete redesign of both the physical and MAC layer. For example, they require to use more than 256 subcarriers in a WiFi channel. On the contrary, REPICK can work under existing 802.11 physical layer with 48 subcarriers.

In [10], Sen et al. designed a contention resolution scheme called T2F. In T2F, in each transmission round, after a two-round frequency domain contention, three nodes are selected and scheduled to transmit packets. This scheduled transmission makes T2F vulnerable to hidden nodes when there are multiple contention domains. However, in REPICK, a time domain transmission retreat scheme is used to ensure that REPICK works well with multiple contention domains. REPICK also differentiate from T2F by employing frequency domain for ACKs. Very recently, in [11], an enhanced version of T2F called Back2F is proposed to handle the case of multiple contention domains. In Back2F, time domain counting down is emulated with subcarrier counting down. However, in Back2F, ACKs are not transmitted together with contention.

Unlike frequency domain contention used in [2] [10] [11], SMACK proposed by Dutta et al. in [5] uses OFDM subcarriers to reduce the overhead of ACKs. SMACK allows multiple nodes to use different subcarriers to send back ACKs for a broadcast message. Different from SMACK, in REPICK, ACKs are piggy-backed with frequency domain contentions. Therefore, the MAC overhead can be further mitigated.

VI. Conclusions

This paper presents REPICK, a random access MAC protocol that can significantly increase throughput in wireless networks. Nodes in REPICK use OFDM subcarriers to perform channel contention before transmission. After the first packet successfully received, a receiver is allowed to take reasonability for the channel contention for follow-up packets. Meanwhile, a piggy-backed ACK is sent back to the sender through activating the sender's identification subcarrier. Through mathematical analysis, we prove that REPICK guarantees a higher throughput compared with 802.11 DCF. With simulation evaluation, we verify our analysis and demonstrate the significant performance gain of REPICK.

Acknowledgment

This research was supported in part by Hong Kong RGC grants 623209, 622410, Huawei-HKUST joint lab, National Natural Science Foundation of China with grants no. as 60933012, 61173156, by a grant from NSFC/RGC under the contract N_HKUST610/11, by grants from HKUST under the contract RPC11EG29, SRFI11EG17-C and SBI09/10.EG01-C, by a grant from Huawei Technologies Co. Ltd. under the contract HUAW18-15L0181011/PN, by a grant from Guangdong Bureau of Science and Technology under the contract GDST11EG06.

References

[1] Z. Zeng, Y. Gao, K. Tan, and P. R. Kumar, "Chain: Introducing minimum controlled coordination into random access mac," in *IEEE INFOCOM* April 2009.

[2] K. Tan, J. Fang, Y. Zhang, S. Chen, L. Shi, J. Zhang, and Y. Zhang, "Fine-grained channel access in wireless lan," in *ACM SIGCOMM*, 2010.

[3] K. Wu, H. Tan, Y. Liu, J. Zhang, Q. Zhang, and L. M. Ni, "Side Channel: Bits over Interference", in *ACM Mobicom* 2010.

[4] K. Wu, H. Tan, H. Ngan, Y. Liu, and L. M. Ni, "Chip Error Pattern Analysis in IEEE 802.15.4", to appear in *IEEE Transactions on Mobile Computing*.

[5] A. Dutta, D. Saha, D. Grunwald, and D. Sicker, "SMACK: a smart acknowledgment scheme for broadcast messages in wireless networks," in *ACM SIGCOMM*, 2009.

[6] F. Cali, M. Conti, and E. Gregori, "Dynamic tuning of the ieee 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Trans. Netw.*, vol. 8, pp. 785-99, December 2000.

[7] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless lans," in *ACM SIGCOMM*, 2005.

[8] X. Yang and N. H. Vaidya, "A wireless mac protocol using implicit pipelining," *IEEE Transactions on Mobile Computing*, vol. 5, pp. 258-73, March 2006.

[9] Y. He, R. Yuan, J. Sun, and W. Gong, "Semi-random backoff: Towards resource reservation for channel access in wireless lans," in *IEEE ICNP*, oct. 2009.

[10] S. Sen, R. R. Choudhury, and S. Nelakuditi, "Listen (on the frequency domain) before you talk," in *ACM HotNets*, 2010.

[11] S. Sen, R. R. Choudhury, and S. Nelakuditi, "No time to countdown: Migrating backoff to the frequency domain," in *ACM MobiCom*, 2011.

[12] M. Jain, J. Choi, T. Kim, D. Bharadia, D. Srinivasan, P. Levis, S. Katti, P. Sinha, S. Seth, "Practical, Real-time Full Duplex Wireless", in *ACM MobiCom*, 2011.

[13] *IEEE* 802.11 - wireless lan medium access control (mac) and physical layer (phy) specifications, IEEE Std. 02.11-2007, June 2007.

[14] *IEEE* 802.11n - enhancements for higher throughput, amendment 4 to ieee 802.11 part 11: Wireless lanmedium access control (mac) and physical layer (phy) specifications, IEEE 802.11n/D3.11, December 2007.

[15] Y. C. Tay and K. C. Chua, "A capacity analysis for the ieee 802.11 mac protocol," *Wirel. Netw.*, vol. 7, pp. 159-71, March 2001.

[16] A. Jayasuriya, S. Perreau, A. Dadej, , and S. Gordon, "Hidden vs. exposed terminal problem in ad hoc networks," in *Australian Telecommunication Networks and Applications Conference*, 2004.