Singletons for Simpletons: Revisiting Windowed Backoff with Chernoff Bounds

Qian M. Zhou
Mississippi State University, Department of Mathematics and Statistics, MS, USA
qz70@msstate.edu

Aiden Calvert
Mississippi School for Mathematics and Science, Columbus, MS, USA
calverta20@themsms.org

Maxwell Young
Mississippi State University, Department of Computer Science and Engineering, MS, USA
myoung@cse.msstate.edu

Abstract

Backoff algorithms are used in many distributed systems where multiple devices contend for a shared resource. For the classic balls-into-bins problem, the number of singletons – those bins with a single ball – is important to the analysis of several backoff algorithms; however, existing analyses employ advanced probabilistic tools to obtain concentration bounds. Here, we show that standard Chernoff bounds can be used instead, and the simplicity of this approach is illustrated by re-analyzing some well-known backoff algorithms.

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases Chernoff bounds, backoff, contention resolution, algorithms

Digital Object Identifier 10.4230/LIPIcs.FUN.2021.24

Funding *Maxwell Young*: This research is supported by the National Science Foundation grant CNS 1816076, and by the U.S. National Institute of Justice (NIJ) Grant 2018-75-CX-K002.

1 Introduction

Backoff algorithms address the general problem of how to share a resource among multiple devices [38]. A ubiquitous application is IEEE 802.11 (WiFi) networks [31, 48, 34], where the resource is a wireless channel, and devices each have packets to send. Any single packet sent uninterrupted over the channel is likely to be received, but if the sending times of two or more packets overlap, communication often fails due to destructive interference at the receiver (i.e., a collision). An important performance metric is the time required for all packets to be sent, which is known as the ***makespan***.

Formal Model. Time is discretized into ***slots***, and each packet can be transmitted within a single slot. Starting from the first slot, a ***batch*** of ***n packets*** is ready to be transmitted on a shared channel. This case, where all packets start at the same time, is sometimes referred to as the ***batched-arrivals*** setting. Each packet can be viewed as originating from a different source device, and going forward we speak only of packets rather than devices.

For any fixed slot, if a single packet sends, then the packet ***succeeds***; however, if two or more packets send, then all corresponding packets ***fail***. A packet that attempts to send in a slot learns whether it succeeded and, if so, the packet takes no further action; otherwise, the packet learns that it failed in that slot, and must try again at a later time.

Background on Analyzing Makespan. A natural question is the following: *For a given backoff algorithm under batched-arrivals, what is the makespan as measured in the number of slots?*

This question was first addressed by Bender et al. [5] who analyze several backoff algorithms that execute over disjoint, consecutive sets of slots called ***windows***. In every window, each packet that has not yet succeeded selects a single slot uniformly at random in which to send. If the packet succeeds, then it leaves the system; otherwise, the failed packet waits for the next window to begin and repeats this process.

Bender et al. [5] analyze several algorithms where windows monotonically increase in size. The well-known ***binary exponential backoff*** algorithm – a critical component of many WiFi standards – exemplifies this behavior, where each successive window increases in size by a factor of 2.[1]

There is a close relationship between the execution of such algorithms in a window, and the popular balls-in-bins scenario, where $\boldsymbol{N}$ balls (corresponding to packets) are dropped uniformly at random into $\boldsymbol{B}$ bins (corresponding to slots). In this context, we are interested in the number of bins containing a single ball, which are sometimes referred to as ***singletons*** [52].

Despite their simple specification, windowed backoff algorithms are surprisingly intricate in their analysis. In particular, obtaining concentration bounds on the number of slots (or bins) that contain a single packet (or ball) – which we will *also* refer to as singletons – is complicated by dependencies that rule out a straightforward application of Chernoff bounds (see Section 2.1). This is unfortunate given that Chernoff bounds are often one of the first powerful probabilistic tools that researchers learn, and they are standard material in a randomized algorithms course.

In contrast, the makespan results in Bender et al. [5] are derived via delay sequences [33, 49], which are arguably a less-common topic of instruction. Alternative tools for handling dependencies include Poisson-based approaches by Mizenmacher [40] and Mitzenmacher and Upfal [39], and the Doob martingale [21], but to the best of our knowledge, these have not been applied to the analysis of windowed backoff algorithms.

1.1 Our Goal

Is there a simpler route to arrive at makespan results for windowed backoff algorithms?

Apart from being a fun theoretical question to explore, an affirmative answer might improve accessibility to the area of backoff algorithms for researchers. More narrowly, this might benefit students embarking on research, many of whom cannot fully appreciate the very algorithms that enable, for example, their ~~Instagram posts~~ access to online course notes.[2] Arguably, Chernoff bounds can be taught without much setup. For example, Dhubashi and Panconesi [21] derive Chernoff bounds starting on page 3, while their discussion of concentration results for dependent variables is deferred until Chapter 5.

What if we could deploy standard Chernoff bounds to analyze singletons? Then, the analysis distills to proving the correctness of a "guess" regarding a recursive formula (a well-known procedure for students) describing the number of packets remaining after each window, and that guess would be accurate with small error probability.

[1] In practice, the doubling terminates at some fixed large value set by the standard.

[2] In our experience, the makespan analysis is inaccessible to most students in the advanced computer networking course.

Finally, while it may not be trivial to show that Chernoff bounds are applicable to backoff, showing that another problem – especially one that has such important applications – succumbs to Chernoff bounds is aesthetically satisfying.

1.2 Results

We show that Chernoff bounds can indeed be used as proposed above. Our approach involves an argument that the indicator random variables for counting singletons satisfy the following property from [22]:

▶ **Property 1.** *Given a set of n indicator random variables $\{X_1, \cdots, X_n\}$, for all subsets $\$ \subset \{1, \cdots, n\}$ the following is true:*

$$Pr\left[\bigwedge_{j\in\$} X_j = 1\right] \leq \prod_{j\in\$} Pr\left[X_j = 1\right]. \tag{1}$$

We prove the following:

▶ **Theorem 1.** *Consider N balls dropped uniformly at random into B bins. Let $I_j = 1$ if bin j contains exactly 1 ball, and $I_j = 0$ otherwise, for $j = 1, \cdots, B$. If $B \geq N + \sqrt{N}$ or $B \leq N - \sqrt{N}$, then $\{I_1, \cdots, I_B\}$ satisfy the Property 1.*

Property 1 permits the use of standard Chernoff bounds; this implication is posed as an exercise by Dubhashi and Panconesi [21] (Problem 1.8), and we provide the argument in our appendix.

We then show how to use Chernoff bounds to obtain asymptotic makespan results for some of the algorithms previously analyzed by Bender et al. [5]: Binary Exponential Backoff (BEB), Fixed Backoff (FB), and Log-Log Backoff (LLB). Additionally, we re-analyze the asymptotically-optimal (non-monotonic) Sawtooth Backoff (STB) from [29, 25].

These algorithms are specified in Section 5, but our makespan results are stated below.

▶ **Theorem 2.** *For a batch of n packets, the following holds with probability at least $1-O(1/n)$:*

- *FB has makespan at most $n \lg\lg n + O(n)$.*
- *BEB has makespan at most $512 n \lg n + O(n)$.*
- *LLB has makespan $O(n \lg\lg n / \lg\lg\lg n)$.*
- *STB has makespan $O(n)$.*

We highlight that both of the cases in Theorem 1, $B \leq N + \sqrt{N}$ and $B \geq N - \sqrt{N}$, are useful. Specifically, the analysis for BEB, FB, and STB uses the first case, while LLB uses both.

1.3 Related Work

Several prior results address dependencies and their relevance to Chernoff bounds and load-balancing in various balls-in-bins scenarios. In terms of backoff, the literature is vast. In both cases, we summarize closely-related works.

Dependencies, Chernoff Bounds, & Ball-in-Bins. Backoff is closely-related to balls-and-bins problems [4, 18, 47, 50], where balls and bins correspond to packets and slots, respectively. Balls-in-bins analysis often arises in problems of load balancing (for examples, see [9, 10, 11]).

Dubhashi and Ranjan [22] prove that the occupancy numbers – random variables N_i denoting the number of balls that fall into bin i – are negatively associated. This result is used by Lenzen and Wattenhofer [35] use it to prove negative association for the random variables that correspond to at most $k \geq 0$ balls.

Czumaj and Stemann [19] examine the maximum load in bins under an adaptive process where each ball is placed into a bin with minimum load of those sampled prior to placement. Negative association of the occupancy numbers is important to this analysis.

Finally, Dubhashi and Ranjan [22] also show that Chernoff bounds remain applicable when the corresponding indicator random variables that are negatively associated. The same result is presented in Dubhashi and Panconesi [21].

Backoff Algorithms. Many early results on backoff are given in the context of statistical queuing-theory (see [30, 28, 43, 26, 30, 27]) where a common assumption is that packet-arrival times are Poisson distributed.

In contrast, for the batched-arrivals setting, the makespan of backoff algorithms with monotonically-increasing window sizes has been analyzed in [5], and with packets of different sizes in [6]. A windowed, but non-monotonic backoff algorithm which is asymptotically optimal in the batched-arrival setting is provided in [25, 29, 2].

A related problem is *contention resolution*, which addresses the time until the first packet succeeds [51, 41, 24, 23]. This has close ties to the well-known problem of leader election (for examples, see [13, 12]).

Several results examine the *dynamic* case where packets arrive over time as scheduled in a worst-case fashion [36, 20, 8]; this is in contrast to batched-arrivals where it is implicitly assumed that the current batch of packets succeeds before the next batch arrives. A similar problem is that of *wake-up* [16, 15, 17, 14, 37, 32], which addresses how long it takes for a single transmission to succeed when packets arrive under the dynamic scenario.

Finally, several results address the case where the shared communication channel is unavailable at due to malicious interference [3, 44, 45, 46, 42, 1, 7].

2 Analysis for Property 1

We present our results on Property 1. Since we believe this result may be useful outside of backoff, our presentation in this section is given in terms of the well-known balls-in-bins terminology, where we have $\boldsymbol{N}$ balls that are dropped uniformly at random into $\boldsymbol{B}$ bins.

2.1 Preliminaries

Throughout, we often employ the following inequalities (see Lemma 3.3 in [46]), and we will refer to the left-hand side (LHS) or right-hand side (RHS) when doing so.

▶ **Fact 1.** *For any $0 < x < 1$, $e^{-x/(1-x)} \leq 1 - x \leq e^{-x}$.*

Knowing that indicator random variables (i.r.v.s) satisfy Property 1 is useful since the following Chernoff bounds can then be applied.

▶ **Theorem 3.** *(Dubhashi and Panconesi [21])*[3] *Let* $X = \sum_i X_i$ *where* $X_1, ..., X_m$ *are i.r.v.s that satisfy Property 1 . For* $0 < \epsilon < 1$*, the following holds:*

$$Pr[X > (1+\epsilon)E[X]] \leq \exp\left(-\frac{\epsilon^2}{3}E[X]\right) \tag{2}$$

$$Pr[X < (1-\epsilon)E[X]] \leq \exp\left(-\frac{\epsilon^2}{2}E[X]\right) \tag{3}$$

We are interested in the i.r.v.s I_j, where:

$$I_j = \begin{cases} 1, & \text{if bin } j \text{ contains exactly 1 ball.} \\ 0, & \text{otherwise.} \end{cases}$$

Unfortunately, there are cases where the I_js fail to satisfy Property 1. For example, consider $N = 2$ balls and $B = 2$ bins. Then, $Pr(I_1 = 1) = Pr(I_2 = 1) = 1/2$, so $Pr(I_1 = 1) \cdot Pr(I_2 = 1) = 1/4$, but $Pr(I_1 = 1 \wedge I_2 = 1) = 1/2$.

A naive approach (although, we have not seen it in the literature) is to leverage the result in [35], that the variables used to count the number of bins with at most k balls are negatively associated. We may bound the number of bins that have at most 1 ball, and the number of bins that have (at most) 0 balls, and then take the difference. However, this is a cumbersome approach, and our result is more direct.

Returning briefly to the context of packets and time slots, another approach is to consider a subtly-different algorithm where a packet sends with probability $1/w$ in each slot of a window with w slots, rather than selecting uniformly at random a single slot to send in. However, as Bender et al. [5] point out, when n is within a constant factor of the window size, there is a constant probability that the packet will not send in *any* slot. Consequently, the number of windows required for all packets to succeed increases by a $\log n$-factor, whereas only $O(\log\log n)$ windows are required under the model used here.

2.2 Property 1 and Bounding Singletons

To prove Theorem 1, we establish the following Lemma 4. For $j = 1, \cdots, B-1$, define:

$$\mathcal{P}_j = Pr\left[I_{j+1} = 1 \mid I_1 = 1, \cdots, I_j = 1\right]$$

which is the conditional probability that bin $j+1$ contains exactly 1 ball given each of the bins $\{1, \cdots, j\}$ contains exactly 1 ball. Note that $Pr[I_j = 1]$ is same for any $j = 1, \cdots, B$, and let:

$$\mathcal{P}_0 \triangleq Pr[I_j = 1] = N\left(\frac{1}{B}\right)\left(1 - \frac{1}{B}\right)^{N-1}. \tag{4}$$

▶ **Lemma 4.** *If* $B \geq N + \sqrt{N}$ *or* $B \leq N - \sqrt{N}$*, the conditional probability* $\mathcal{P}_j$ *is a monotonically non-increasing function of* j*, i.e.,* $\mathcal{P}_j \geq \mathcal{P}_{j+1}$*, for* $j = 0, \cdots, B-2$.

Proof. First, for $j = 1, \cdots, \min\{B, N\} - 1$, the conditional probability can be expressed as

$$\mathcal{P}_j = (N-j)\left(\frac{1}{B-j}\right)\left(1 - \frac{1}{B-j}\right)^{N-j-1}. \tag{5}$$

[3] This is stated in Problem 1.8 in [21]; we present a proof in Section A of our appendix.

Note that $\mathcal{P}_0$ in (4) is equal to (5) with $j = 0$.

For $B \geq N + \sqrt{N}$, we note that beyond the range $j = 1, ..., , \min\{B, N\} - 1$ (i.e., $N - 1$), it must be that $\mathcal{P}_j = 0$. In other words, $\mathcal{P}_j = 0$ for $j = N, N + 1, \cdots , B - 1$ since all balls have already been placed. Thus, we need to prove $\mathcal{P}_j \geq \mathcal{P}_{j+1}$, for $j = 0, \cdots , N - 2$.

On the other hand, if $B \leq N - \sqrt{N}$, we need to prove $\mathcal{P}_j \geq \mathcal{P}_{j+1}$, for $j = 0, \cdots , B - 2$. Thus, this lemma is equivalent to prove if $B \geq N + \sqrt{N}$ or $B \leq N - \sqrt{N}$, the ratio $\mathcal{P}_j/\mathcal{P}_{j+1} \geq 1$, for $j = 0, \cdots , \min\{B, N\} - 2$.

Using the expression (5), the ratio can be expressed as

$$\begin{aligned}
\frac{\mathcal{P}_j}{\mathcal{P}_{j+1}} &= \frac{(N-j)\left(\frac{1}{B-j}\right)\left(1-\frac{1}{B-j}\right)^{N-j-1}}{(N-j-1)\left(\frac{1}{B-j-1}\right)\left(1-\frac{1}{B-j-1}\right)^{N-j-2}} \\
&= \frac{1}{\left(\frac{B-j}{N-j}\right)\left(\frac{N-j-1}{B-j-1}\right)} \cdot \frac{\left(1-\frac{1}{B-j}\right)^{N-j-1}}{\left(1-\frac{1}{B-j-1}\right)^{N-j-2}} \\
&= \frac{1}{\left(\frac{B-j}{N-j}\right)\left(\frac{N-j-1}{B-j-1}\right)} \cdot \frac{\left(\frac{B-j-1}{B-j}\right)^{N-j-1}}{\left(\frac{B-j-2}{B-j-1}\right)^{N-j-1}\left(\frac{B-j-1}{B-j-2}\right)} \\
&= \frac{1}{\left(\frac{B-j}{N-j}\right)\left(\frac{N-j-1}{B-j-2}\right)} \cdot \left(\frac{\frac{B-j-1}{B-j}}{\frac{B-j-2}{B-j-1}}\right)^{N-j-1} \\
&= \frac{\left(1+\frac{1}{(B-j)(B-j-2)}\right)^{N-j-1}}{\frac{(N-j-1)(B-j)}{(N-j)(B-j-2)}}.
\end{aligned}$$

Let $a = N - j$, then $2 \leq a \leq N$; and let $y = B - N$. Thus, the ratio becomes

$$\frac{\mathcal{P}_j}{\mathcal{P}_{j+1}} = \frac{\left[1+\frac{1}{(a+y)(a+y-2)}\right]^{a-1}}{\frac{(a-1)(a+y)}{a(a+y-2)}}.$$

By the Binomial theorem, we have

$$\left[1+\frac{1}{(a+y)(a+y-2)}\right]^{a-1} = 1+\frac{a-1}{(a+y)(a+y-2)}+\sum_{k=2}^{a-1}\binom{a-1}{k}\left[\frac{1}{(a+y)(a+y-2)}\right]^{k}.$$

Thus, the ratio can be written as:

$$\begin{aligned}
\frac{\mathcal{P}_j}{\mathcal{P}_{j+1}} &= \frac{a(a+y-2)}{(a-1)(a+y)} + \frac{a}{(a+y)^2} + \frac{\sum_{k=2}^{a-1}\binom{a-1}{k}\left[\frac{1}{(a+y)(a+y-2)}\right]^{k}}{\frac{(a-1)(a+y)}{a(a+y-2)}} \\
&= \frac{a^3+2a^2y-a^2+ay^2-2ay-a}{a^3+2a^2y-a^2+ay^2-2ay-y^2} + \frac{\sum_{k=2}^{a-1}\binom{a-1}{k}\left[\frac{1}{(a+y)(a+y-2)}\right]^{k}}{\frac{(a-1)(a+y)}{a(a+y-2)}} \\
&= \frac{a^3+2a^2y-a^2+ay^2-2ay-a+(y^2-y^2)}{a^3+2a^2y-a^2+ay^2-2ay-y^2} + \frac{\sum_{k=2}^{a-1}\binom{a-1}{k}\left[\frac{1}{(a+y)(a+y-2)}\right]^{k}}{\frac{(a-1)(a+y)}{a(a+y-2)}} \\
&= 1+\frac{y^2-a}{(a+y)^2(a-1)} + \frac{\sum_{k=2}^{a-1}\binom{a-1}{k}\left[\frac{1}{(a+y)(a+y-2)}\right]^{k}}{\frac{(a-1)(a+y)}{a(a+y-2)}}. \qquad (6)
\end{aligned}$$

Note that because $0 \leq j \leq \min\{B, N\} - 2$, then $a + y = B - j \geq 2$. Thus, the third term in (6) is always non-negative. If $y = B - N \geq \sqrt{N}$ or $y \leq -\sqrt{N}$, then $y^2 \geq N \geq a$ for any $2 \leq a \leq N$. Consequently, the ratio $\mathcal{P}_j / \mathcal{P}_{j+1} \geq 1$. ◀

We can now give our main argument:

Proof of Theorem 1. Let s denote the size of the subset $\$ \subset \{1, \cdots, B\}$, i.e. the number of bins in $\$$. First, note that if $B \geq N + \sqrt{N}$, when $s > N$ (i.e., more bins than balls), the probability on the left hand side (LHS) of (1) is 0, thus, the inequality (1) holds. In addition, shown above $Pr[I_j = 1] = \mathcal{P}_0$ for any $j = 1, \cdots, B$. Thus, the right hand side of (1) becomes $\mathcal{P}_0^s$. Thus, we need to prove for any subset, denoted as $\$ = \{j_1, \cdots, j_s\}$ with $1 \leq s \leq \min\{B, N\}$

$$Pr\left[\bigwedge_{k=1}^{s} I_{j_k} = 1\right] \leq \mathcal{P}_0^s.$$

The LHS can be written as:

$$\begin{aligned} &= \quad Pr\left[I_{j_s} = 1 \mid \bigwedge_{k=1}^{s-1} I_{j_k} = 1\right] Pr\left[\bigwedge_{k=1}^{s-1} I_{j_k} = 1\right] \\ &= \mathcal{P}_{s-1} Pr\left[\bigwedge_{k=1}^{s-1} I_{j_k} = 1\right] \end{aligned}$$

$$\begin{aligned} &= \mathcal{P}_{s-1} Pr\left[I_{j_{s-1}} = 1 \mid \bigwedge_{k=1}^{s-2} I_{j_k} = 1\right] Pr\left[\bigwedge_{k=1}^{s-2} I_{j_k} = 1\right] \\ &= \mathcal{P}_{s-1}\mathcal{P}_{s-2} Pr\left[\bigwedge_{k=1}^{s-2} I_{j_k} = 1\right] \\ &\vdots \\ &= \mathcal{P}_{s-1}\mathcal{P}_{s-2} \cdots \mathcal{P}_0 \end{aligned}$$

Lemma 4 shows that if $B \geq N + \sqrt{N}$ or $B \leq N - \sqrt{N}$, $\mathcal{P}_j$ is a non-increasing function of $j = 0, \cdots, B - 1$. Consequently, $\mathcal{P}_0 \geq \mathcal{P}_j$, for $j = 1, \cdots, B - 1$. Thus:

$$Pr\left[\bigwedge_{k=1}^{s} I_{j_k} = 1\right] \leq \mathcal{P}_0^s,$$

and so the bound in Equation (1) holds. ◀

The standard Cheroff bounds of Theorem 3 now apply, and we use them obtain bounds on the number of singletons. For ease of presentation, we occasionally use $\exp(x)$ to denote e^x.

▶ **Lemma 5.** *For N balls that are dropped into B bins where $B \geq N + \sqrt{N}$ or $B \leq N - \sqrt{N}$, the following is true for any $0 < \epsilon < 1$.*

- *The number of singletons is at least $\frac{(1-\epsilon)N}{e^{N/(B-1)}}$ with probability at least $1 - e^{\frac{-\epsilon^2 N}{2\exp(N/(B-1))}}$.*
- *The number of singletons is at most $\frac{(1+\epsilon)N}{e^{(N-1)/B}}$ with probability at least $1 - e^{\frac{-\epsilon^2 N}{3\exp(N/(B-1))}}$.*

Proof. We begin by calculating the expected number of singletons. Let I_i be an indicator random variable such that $I_i = 1$ if bin i contains a single ball; otherwise, $I_i = 0$. Note that:

$$\begin{aligned} Pr(I_i = 1) &= \binom{N}{1}\left(\frac{1}{B}\right)\left(1 - \frac{1}{B}\right)^{N-1} \\ &\geq \binom{N}{1}\left(\frac{1}{B}\right)\left(1 - \frac{1}{B}\right)^{N} \\ &\geq \frac{N}{Be^{(N/(B-1))}} \end{aligned} \tag{7}$$

where the last line follows from the LHS of Fact 1. Let $I = \sum_{i=1}^{B} I_i$ be the number of singletons. We have:

$$\begin{aligned} E[I] &= \sum_{i=1}^{B} E[I_i] \quad \text{by linearity of expectation} \\ &\geq \frac{N}{e^{(N/(B-1))}} \quad \text{by Equation (7)} \end{aligned}$$

Next, we derive a concentration result around this expected value. Since $B \geq N + \sqrt{N}$ or $B \leq N - \sqrt{N}$, Theorem 1 guarantees that the I_is are negatively associated, and we may apply the Chernoff bound in Equation 3 to obtain:

$$Pr\left(I < (1-\epsilon)\frac{N}{e^{(N/(B-1))}}\right) \leq \exp\left(-\frac{\epsilon^2 N}{2e^{(N/(B-1))}}\right)$$

which completes the lower-bound argument. The upper bound is nearly identical. ◀

3 Bounding Remaining Packets

In this section, we derive tools for bounding the number of packets that remain as we progress from one window to the next.

All of our results hold for sufficiently large $n > 0$. Let $\boldsymbol{w_i}$ denote the number of slots in window $i \geq 0$. Let $\boldsymbol{m_i}$ be the number of packets at the start of window $i \geq 0$.

We index windows starting from 0, but this does not necessarily correspond to the initial window executed by a backoff algorithm. Rather, in our analysis, window 0 corresponds to the first window where packets start to succeed in large numbers; this is different for different backoff algorithms.

For example, BEB's initial window consists of a single slot, and does not play an important role in the makespan analysis. Instead, we apply Chernoff bounds once the window size is at least $n+\sqrt{n}$, and this corresponds to window 0. In contrast, for FB, the first window (indeed, *each* window) has size $\Theta(n)$, and window 0 is indeed this first window for our analysis. This indexing is useful for our inductive arguments presented in Section 4.

3.1 Analysis

Our method for upper-bounding the makespan operates in three stages. First, we apply an inductive argument – employing Case 1 in Corollary 6 below – to cut down the number of packets from n to less than $n^{0.7}$. Second, Case 2 of Corollary 6 is used whittle the remaining packets down to $O(n^{0.4})$. Third, we hit the remaining packets with a constant number of calls to Lemma 7; this is the essence of Lemma 8.

Intuition for Our Approach. There are a couple things worth noting. To begin, why not carry the inductive argument further to reduce the number of packets to $O(n^{0.4})$ directly (i.e., skip the second step above)? Informally, our later inductive arguments show that m_{i+1} is roughly at most $n/2^{2^i}$, and so $i \approx \lg\lg(n)$ windows should be sufficient. However, $\lg\lg(n)$ is not necessarily an integer and we may need to take its floor. Given the double exponential, taking the floor (subtracting 1) results in $m_{i+1} \geq \sqrt{n}$. Therefore, the equivalent of our second step will still be required. Our choice of $n^{0.7}$ is not the tightest, but it is chosen for simpicity.

The second threshold of $O(n^{0.4})$ is also not completely arbitrary. In the (common) case where $w_0 \geq n + \sqrt{n}$, note that we require $O(n^{1/2-\delta})$ packets remaining, for some constant $\delta > 0$, in order to get a useful bound from Lemma 7. It is possible that after the inductive argument, that this is already satisfied; however, if not, then Case 2 of Corollary 6 enforces this. Again, $O(n^{0.4})$ is chosen for ease of presentation; there is some slack.

▶ **Corollary 6.** *For $w_i \geq n + \sqrt{n}$, the following is true with probability at least $1 - 1/n^2$:*

- *Case 1. If $m_i \geq n^{7/10}$, then $m_{i+1} < \frac{(5/4)m_i^2}{n}$.*
- *Case 2. If $n^{0.4} \leq m_i < n^{7/10}$, then $m_{i+1} = O(n^{2/5})$.*

Proof. For Case 1, we apply the first result of Lemma 5 with $\epsilon = \frac{\sqrt{4e\ln n}}{n^{1/3}}$, which implies with probability at least $1 - \exp(-\frac{4e\ln n}{n^{2/3}}\frac{n^{0.7}}{2}) \geq 1 - \exp(-2\ln n) \geq 1 - 1/n^2$:

$$\begin{aligned}
m_{i+1} &\leq m_i - \frac{(1-\epsilon)m_i}{e^{m_i/(w_i-1)}} \\
&\leq m_i\left(1 - \frac{1}{e^{m_i/(w_i-1)}} + \epsilon\right) \\
&\leq m_i\left(\frac{m_i}{w_i - 1} + \epsilon\right) \quad \text{by RHS of Fact 1} \\
&\leq \frac{m_i^2}{n} + m_i\epsilon \quad \text{since } w_i \geq n + \sqrt{n} \qquad (8) \\
&\leq \frac{m_i^2}{n} + \left(\frac{m_i}{n^{1/3}}\right)\sqrt{4e\ln n} \\
&< \frac{(5/4)m_i^2}{n} \quad \text{since } m_i \geq n^{7/10}
\end{aligned}$$

where 5/4 is chosen for ease of presentation.

For Case 2, we again apply the first result of Lemma 5, but with $\epsilon = \sqrt{\frac{4e\ln n}{m}}$. Then, with probability at least $1 - 1/n^2$, the first and second terms in Equation 8 are at most $n^{0.4}$ and $O(n^{0.35}\sqrt{\ln n})$, respectively, for the any $n^{0.4} \leq m_i \leq n^{7/10}$. ◀

The following lemma is useful for achieving a with-high-probability guarantee when the number of balls is small relative to the number of bins.

▶ **Lemma 7.** *Assume $w_i > 2m_i$. With probability at least $1 - \frac{m_i^2}{w_i}$, all packets succeed in window i.*

Proof. Consider placements of packets in the window that yield at most one packet per slot. Note that once a packet is placed in a slot, there is one less slot available for each remaining packet yet to be placed. Therefore, there are $w_i(w_i - 1)\cdots(w_i - m_i + 1)$ such placements.

Since there are $w_i^{m_i}$ ways to place m_i packets in w_i slots, it follows that the probability that each of the m_i packets chooses a different slot is:

$$\frac{w_i(w_i-1)\cdots(w_i-m_i+1)}{w_i^{m_i}}.$$

We can lower bound this probability:

$$\begin{aligned}
&= \frac{w_i^{m_i}(1-1/w_i)\cdots(1-(m_i-1)/w_i)}{w_i^{m_i}} \\
&\geq e^{-\sum_{j=1}^{m_i-1}\frac{j}{w_i-j}} \quad \text{by LHS of Fact 1} \\
&\geq e^{-\sum_{j=1}^{m_i-1}\frac{2j}{w_i}} \quad \text{since } w_i > 2m_i > 2j \text{ which} \\
&\qquad\qquad\qquad\quad \text{leads to } \tfrac{j}{w_i-j} < \tfrac{2j}{w_j} \\
&= e^{-(1/w_i)(m_i-1)m_i} \quad \text{by sum of natural numbers} \\
&\geq 1 - \frac{m_i^2}{w_i} + \frac{m_i}{w_i} \quad \text{by RHS of Fact 1} \\
&> 1 - \frac{m_i^2}{w_i}
\end{aligned}$$

as claimed. ◀

▶ **Lemma 8.** *Assume a batch of $m_i < n^{7/10}$ packets that execute over a window of size w_i, where $w_i \geq n + \sqrt{n}$ for all i. Then, with probability at least $1 - O(1/n)$, any monotonic backoff algorithm requires at most 6 additional windows for all remaining packets to succeed.*

Proof. If $m_i \geq n^{0.4}$, then Case 2 of Corollary 6 implies $m_{i+1} = O(n^{0.4})$; else, we do not need to invoke this case. By Lemma 7, the probability that any packets remain by the end of window $i+1$ is $O(n^{0.8}/n) = O(1/n^{0.2})$; refer to this as the probability of failure. Subsequent windows increase in size monotonically, while the number of remaining packets decreases monotonically. Therefore, the probability of failure is $O(1/n^{0.2})$ in any subsequent window, and the probability of failing over all of the next 5 windows is less than $O(1/n)$. It follows that at most 6 windows are needed for all packets to succeed. ◀

4 Inductive Arguments

We present two inductive arguments for establishing upper bounds on m_i. Later in Section 5, these results are leveraged in our makespan analysis, and extracting them here allows us to modularize our presentation. Lemma 9 applies to FB, BEB, and LLB, while Lemma 10 applies to STB. We highlight that a single inductive argument would suffice for all algorithms – allowing for a simpler presentation – if we only cared about asymptotic makespan. However, in the case of FB we wish to obtain a tight bound on the first-order term, which is one of the contributions in [5].

In the following, we specify $m_0 \leq n$ since a (very) few packets may have succeeded prior to window 0; recall, this is the window where a large number of packets are expected to succeed.

▶ **Lemma 9.** *Consider a batch of $m_0 \leq n$ packets that execute over windows $w_i \geq m_0 + \sqrt{m_0}$ for all $i \geq 0$. If $m_i \geq n^{7/10}$, then $m_{i+1} \leq (4/5)\frac{m_0}{2^{2^i \lg(5/4)}}$ with error probability at most $(i+1)/n^2$.*

Proof. We argue by induction on $i \geq 0$.

Base Case. Let $i = 0$. Using Lemma 5:

$$\begin{aligned} m_1 &\leq m_0 - \frac{(1-\epsilon)m_0}{e^{m_0/(w_0-1)}} \\ &\leq m_0\left(1 - \frac{1}{e^{m_0/(w_0-1)}} + \epsilon\right) \end{aligned}$$

$$\begin{aligned} &\leq m_0\left(1 - \frac{1}{e} + \epsilon\right) \\ &\leq (0.64)m_0 \end{aligned}$$

where the last line follows by setting $\epsilon = \frac{\sqrt{4e\ln n}}{n^{1/3}}$, and assuming n is sufficiently large to satisfy the inequality; this gives an error probability of at most $1/n^2$. The base case is satisfied since $(4/5)\frac{m_0}{2^{2^i \lg(5/4)}} = (0.64)m_0$.

Induction Hypothesis (IH). For $i \geq 1$, assume $m_i \leq (4/5)\frac{m_0}{2^{2^{i-1}\lg(5/4)}}$ with error probability at most i/n^2.

Induction Step. For window $i \geq 1$, we wish to show that $m_{i+1} \leq (4/5)\frac{m_0}{2^{2^i \lg(5/4)}}$ with an error bound of $(i+1)/n^2$. Addressing the number of packets, we have:

$$\begin{aligned} m_{i+1} &\leq \frac{(5/4)m_i^2}{w_i} \\ &\leq \left(\frac{4\,m_0}{5\cdot 2^{2^{i-1}\lg(5/4)}}\right)^2\left(\frac{5}{4w_i}\right) \\ &\leq \left(\frac{4m_0}{5\cdot 2^{2^i \lg(5/4)}}\right)\left(\frac{m_0}{w_i}\right) \\ &< \left(\frac{4m_0}{5\cdot 2^{2^i \lg(5/4)}}\right) \text{since } w_i > n \end{aligned}$$

The first line follows from Case 1 of Corollary 6, which we may invoke since $w_i \geq m_0 + \sqrt{m_0}$ for all $i \geq 0$, and $m_i \geq n^{7/10}$ by assumption. This yields an error of at most $1/n^2$, and so the total error is at most $i/n^2 + 1/n^2 = (i+1)/n^2$ as desired. The second line follows from the IH. ◀

A nearly identical lemma is useful for upper-bounding the makespan of STB. The main difference arises from addressing the decreasing window sizes in a run, and this necessitates the condition that $w_i \geq m_i + \sqrt{m_i}$ rather than $w_i \geq m_0 + \sqrt{m_0}$ for all $i \geq 0$. Later in Section 5, we start analyzing STB when the window size reaches $4n$; this motivates the condition that $w_i \geq 4n/2^i$ our next lemma.

▶ **Lemma 10.** *Consider a batch of $m_0 \leq n$ packets that execute over windows of size $w_i \geq m_i + \sqrt{m_i}$ and $w_i \geq 4n/2^i$ for all $i \geq 0$. If $m_i \geq n^{7/10}$, then $m_{i+1} \leq (4/5)\frac{m_0}{2^i 2^{2^i \lg(5/4)}}$ with error probability at most $(i+1)/n^2$.*

Proof. We argue by induction on $i \geq 0$.

Base Case. Nearly identical to the base case in proof of Lemma 9; note the bound on m_{i+1} is identical for $i = 0$.

Induction Hypothesis (IH). For $i \geq 1$, assume $m_i \leq (4/5)\frac{m_0}{2^{i-1}2^{2^{i-1}\lg(5/4)}}$ with error probability at most i/n^2.

Induction Step. For window $i \geq 1$, we wish to show that $m_{i+1} \leq (4/5)\frac{m_0}{2^i 2^{2^i \lg(5/4)}}$ with an error bound of $(i+1)/n^2$ (we use the same ϵ as in Lemma 9). Addressing the number of packets, we have:

$$\begin{aligned} m_{i+1} &\leq \frac{(5/4)m_i^2}{w_i} \\ &\leq \left(\frac{4m_0}{5 \cdot 2^{i-1}2^{2^{i-1}\lg(5/4)}}\right)^2 \left(\frac{5}{4w_i}\right) \\ &\leq \left(\frac{4m_0}{5 \cdot 2^i 2^{2^i \lg(5/4)}}\right)\left(\frac{m_0}{2^{i-2}w_i}\right) \\ &\leq \left(\frac{4m_0}{5 \cdot 2^i 2^{2^i \lg(5/4)}}\right) \text{ since } w_i \geq 4n/2^i \end{aligned}$$

Again, the first line follows from Case 1 of Corollary 6, which we may invoke since $w_i \geq m_0 + \sqrt{m_0}$ for all $i \geq 0$, and $m_i \geq n^{7/10}$ by assumption. This gives the desired error bound of $i/n^2 + 1/n^2 = (i+1)/n^2$. The second line follows from the IH. ◀

5 Bounding Makespan

We begin by describing the windowed backoff algorithms Fixed Backoff (FB), Binary Exponential Backoff (BEB), and Log-Log Backoff (LLB) analyzed in [5]. Recall that, in each window, a packet selects a single slot uniformly at random to send in. Therefore, we need only specify how the size of successive windows change.

FB is the simplest, with all windows having size $\Theta(n)$. The value of hidden constant does not appear to be explicitly specified in the literature, but we observe that Bender et al. [5] use $3e^3$ in their upper-bound analysis. Here, we succeed using a smaller constant; namely, any value at least $1 + 1/\sqrt{n}$.

BEB has an initial window size of 1, and each successive window doubles in size.

LLB has an initial window size of 2, and for a current window size of w_i, it executes $\lceil \lg\lg(w_i) \rceil$ windows of that size before doubling; we call these sequence of same-sized windows a ***plateau***.[4]

STB is non-monotonic and executes over a doubly-nested loop. The outer loop sets the current window size w to be double that used in the preceding outer loop and each packet selects a single slot to send in; this is like BEB. Additionally, for each such w, the inner loop executes over $\lg w$ windows of decreasing size: $w, w/2, w/4, ..., 1$; this sequence of windows is referred to as a ***run***. For each window in a run, a packet chooses a slot uniformly at random in which to send.

5.1 Analysis

The following results employ tools from the prior sections a constant number of times, and each tool has error probability either $O(\log n/n^2)$ or $O(\frac{1}{n})$. Therefore, all following theorems hold with probability at least $1 - O(1/n)$, and we omit further discussion of error.

[4] As stated by Bender et al. [5], an equivalent (in terms of makespan) specification of LLB is that $w_{i+1} = (1 + 1/\lg\lg(w_i))w_i$.

▶ **Theorem 11.** *The makespan of FB with window size at least $n+\sqrt{n}$ is at most $n\lg\lg n + O(n)$ and at least $n\lg\lg n - O(n)$.*

Proof. Since $w_i \geq n+\sqrt{n}$ for all $i \geq 0$, by Lemma 9 less than $n^{7/10}$ packets remain after $\lg\lg(n)+1$ windows; to see this, solve for i in $(4/5)\frac{n}{2^{2^i \lg(5/4)}} = n^{0.7}$. By Lemma 8, all remaining packets succeed within 6 more windows. The corresponding number of slots is $(\lg\lg n + 7)(n+\sqrt{n}) = n\lg\lg n + O(n)$. ◀

▶ **Theorem 12.** *The makespan of BEB is at most $512n\lg n + O(n)$.*

Proof. Let W be the first window of size at least $n+\sqrt{n}$ (and less than $2(n+\sqrt{n})$). Assume no packets finish before the start of W; otherwise, this can only improve the makespan. By Lemma 9 less than $n^{7/10}$ packets remain after $\lg\lg(n)+1$ windows. By Lemma 8 all remaining packets succeed within 6 more windows. Since W has size less than $2(n+\sqrt{n})$, the number of slots until the end of W, plus those for the $\lg\lg(n)+7$ subsequent windows, is less than:

$$\begin{aligned} &\left(\sum_{j=0}^{\lg(2(n+\sqrt{n}))} 2^j\right) + \left(\sum_{k=1}^{\lg\lg(n)+7} 2(n+\sqrt{n})2^k\right) \\ = \quad & 512(n+\sqrt{n})\lg n + O(n) \end{aligned}$$

by the sum of a geometric series. ◀

▶ **Theorem 13.** *The makespan of STB is $O(n)$.*

Proof. Let W be the first window of size at least $4n$. Assume no packets finish before the start of W, that is $m_0 = n$; else, this can only improve the makespan.

While $m_i \geq n^{0.7}$, our analysis examines the windows in the run starting with window W, and so $w_0 \geq 4n, w_1 \geq 2n$, etc. To invoke Lemma 10, we must ensure that the condition $w_i \geq m_i + \sqrt{m_i}$ holds in each window of this run. This holds for $i = 0$, since $w_0 = 4n \geq n+\sqrt{n}$.

For $i \geq 1$, we argue this inductively by proving $m_i \leq (5/4)^{2^{i-1}-1}\frac{n}{3^{2^i-1}}$. For the base case $i = 1$, Lemma 5 implies that $m_1 \leq n(1-e^{-n/(4n-1)}+\epsilon) \leq n(1-e^{-1/3}+\epsilon) \leq n/3$, where ϵ is given in Lemma 6. For the inductive step, assume that $m_i \leq (5/4)^{2^{i-1}-1}\frac{n}{3^{2^{i-1}}}$ for all $i \geq 2$. Then, by Case 1 of Corollary 6:

$$\begin{aligned} m_{i+1} &\leq (5/4)m_i^2/n \\ &\leq (5/4)\left((5/4)^{2^{i-1}-1}\frac{n}{3^{2^{i-1}}}\right)^2/n \\ &\leq (5/4)^{2^i-1}\frac{n}{3^{2^i}} \end{aligned}$$

where the second line follows from the assumption, and so the inductive step holds. On the other hand, at window i, $w_i \geq \frac{4n}{2^i} > \frac{4n}{(5/2)\cdot(12/5)^{2^{i-1}}} = 2\cdot(5/4)^{2^{i-1}-1}\frac{n}{3^{2^{i-1}}} \geq 2m_i > m_i + \sqrt{m_i}$ holds.

Lemma 10 implies that after $\lg\lg n + O(1)$ windows in this run, less than $n^{0.7}$ packets remain. Pessimistically, assume no other packets finish in the run. The next run starts with a window of size at least $8n$, and by Lemma 8, all remaining packets succeed within the first 6 windows of this run.

We have shown that STB terminates within at most $\lceil\lg(n)\rceil + O(1)$ runs. The total number of slots over all of these runs is $O(n)$ by a geometric series. ◀

It is worth noting that STB has asymptotically-optimal makespan since we cannot hope to finish n packets in $o(n)$ slots.

Bender et al. [5] show that the optimal makespan for any *monotonic* windowed backoff algorithm is $O(n \lg\lg n / \lg\lg\lg n)$ and that LLB achieves this. We re-derive the makespan for LLB.

▶ **Theorem 14.** *The makespan of LLB is* $O\left(\frac{n \lg\lg n}{\lg\lg\lg n}\right)$.

Proof. For the first part of our analysis, assume $n/\ln\ln\ln n \le m_0 \le n$ packets remain. Consider the first window with size $w_0 = cn/\ln\ln\ln n$ for some constant $c \ge 8$. By Lemma 5, each window finishes at least the following number of packets:

$$\begin{aligned}
\frac{(1-\epsilon)m_0}{e^{\frac{m_0}{(cn/\ln\ln\ln n)-1}}} &> \frac{(1-\epsilon)n}{e^{\frac{n}{(cn/\ln\ln\ln n)-1}} \cdot \ln\ln\ln n} \\
&= \frac{(1-\epsilon)n}{(\ln\ln n)^{\frac{2}{c}} \cdot \ln\ln\ln n} \\
&= \frac{(1-\epsilon)n}{(\ln\ln n)^{\frac{\ln\ln\ln\ln n}{\ln\ln\ln n} + \frac{2}{c}}} \\
&> \frac{n}{(\ln\ln n)^{\frac{3}{c}}}
\end{aligned}$$

where the third line follows from noting that $(\ln\ln n)^{\ln(\ln\ln\ln n)} = (\ln\ln\ln n)^{\ln(\ln\ln n)}$, and the last line follows for sufficiently-large n. Setting $\epsilon = \sqrt{\frac{4e\ln^2(n)}{n}}$ suffices to give an error probability at most $\exp(-\frac{4e\ln^2(n)}{n} \cdot \frac{n}{2\ln\ln\ln(n) e^{\frac{n}{(cn/\ln\ln\ln n)-1}}}) \le 1/n^2$.

Observe that in this first part of the analysis, we rely on $w_i \le m_i - \sqrt{m_i}$ or $w_i \ge m_i + \sqrt{m_i}$ in order to apply Lemma 5. However, after enough packets succeed, neither of these inequalities may hold. But there will be at most a single plateau with windows of size $O(n/\ln\ln\ln n)$ where this occurs, since the window size will then double. During this plateau, which consists of $O(\lg\lg(n/\ln\ln\ln n)) = O(\lg\lg n)$ windows, we pessimistically assume no packets succeed.

Therefore, starting with n packets, after at most $\frac{n - n/\ln\ln\ln n}{n/(\ln\ln n)^{3/c}} + O(\lg\lg n) = O(\ln\ln n)$ windows, the number of remaining packets is less than $n/\ln\ln\ln n$, and the first part of our analysis is over.

Over the next two plateaus, LLB has at least $2\lg\lg(n) - O(1)$ windows of size $\Theta(n/\ln\ln\ln n)$. Since in this part of the analysis, $w_i \ge 8n/\ln\ln\ln n$ and $m_i < n/\ln\ln\ln n$, we have $w_i \ge m_i + \sqrt{m_i}$. Therefore, we may invoke Lemma 9, which implies that after at most $\lg\lg(n) + 1$ windows, less than $n^{0.7}$ packets remain. If at least $n^{2/5}$ packets still remain, by Case 2 of Corollary 1, at most $O(n^{2/5})$ packets remain by the end of the next window, and they will finish within an additional 6 windows by Lemma 8.

Finally, tallying up over both parts of the analysis, the makespan is $O(\ln\ln n)O(\frac{n}{\ln\ln\ln n}) = O(\frac{n\ln\ln n}{\ln\ln\ln n})$. ◀

6 Discussion

We have argued that standard Chernoff bounds can be applied to analyze singletons, and we illustrate how they simplify the analysis of several backoff algorithms under batched arrivals.

While our goal was only to demonstrate the benefits of this approach, natural extensions include the following. First, there is some slack in our arguments, and we can likely derive tighter constants in our analysis. For example, the number of windows required in Lemma 8 might be reduced; this would reduce the leading constant for our BEB analysis.

Second, we strongly believe that lower bounds can be proved using this approach. In fact, Max bets Qian (under penalty of eating bitter melon) that a lower bound on FB of $n \lg \lg n - O(n)$ can be proved, which is tight in the highest-order term.

Third, a similar treatment is possible for polynomial backoff or generalized exponential backoff (see [5] for the specification of these algorithms).

Fourth, a plausible next step is to examine whether we can extend this type of analysis to the case where packets have different sizes, as examined in [6].

References

1 Lakshmi Anantharamu, Bogdan S. Chlebus, Dariusz R. Kowalski, and Mariusz A. Rokicki. Medium access control for adversarial channels with jamming. In *Proceedings of the* 18^{th} *International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, pages 89–100, 2011.

2 Antonio Fernández Anta, Miguel A. Mosteiro, and Jorge Ramón Muñoz. Unbounded contention resolution in multiple-access channels. *Algorithmica*, 67(3):295–314, 2013.

3 Baruch Awerbuch, Andrea Richa, and Christian Scheideler. A jamming-resistant MAC protocol for single-hop wireless networks. In *Proceedings of the 27th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 45–54, 2008.

4 Yossi Azar, Andrei Z. Broder, Anna R. Karlin, and Eli Upfal. Balanced allocations. *SIAM J. Comput.*, 29(1):180–200, September 1999.

5 Michael A. Bender, Martin Farach-Colton, Simai He, Bradley C. Kuszmaul, and Charles E. Leiserson. Adversarial contention resolution for simple channels. In *Proceedings of the 17th Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 325–332, 2005.

6 Michael A. Bender, Jeremy T. Fineman, and Seth Gilbert. Contention Resolution with Heterogeneous Job Sizes. In *Proceedings of the 14th Conference on Annual European Symposium (ESA)*, pages 112–123, 2006.

7 Michael A. Bender, Jeremy T. Fineman, Seth Gilbert, and Maxwell Young. How to scale exponential backoff: Constant throughput, polylog access attempts, and robustness. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2016.

8 Michael A. Bender, Tsvi Kopelowitz, Seth Pettie, and Maxwell Young. Contention resolution with log-logstar channel accesses. In *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing*, STOC '16, pages 499–508, 2016.

9 Petra Berenbrink, Artur Czumaj, Matthias Englert, Tom Friedetzky, and Lars Nagel. Multiple-choice balanced allocation in (almost) parallel. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX-RANDOM)*, pages 411–422, 2012.

10 Petra Berenbrink, Artur Czumaj, Angelika Steger, and Berthold Vöcking. Balanced allocations: The heavily loaded case. *SIAM J. Comput.*, 35(6):1350–1385, 2006.

11 Petra Berenbrink, Kamyar Khodamoradi, Thomas Sauerwald, and Alexandre Stauffer. Balls-into-bins with nearly optimal load distribution. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 326–335, 2013.

12 Yi-Jun Chang, Varsha Dani, Thomas P. Hayes, Qizheng He, Wenzheng Li, and Seth Pettie. The energy complexity of broadcast. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*, PODC '18, pages 95–104, 2018.

13 Yi-Jun Chang, Tsvi Kopelowitz, Seth Pettie, Ruosong Wang, and Wei Zhan. Exponential separations in the energy complexity of leader election. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 771–783, 2017.

14 Bogdan S. Chlebus, Gianluca De Marco, and Dariusz R. Kowalski. Scalable wake-up of multi-channel single-hop radio networks. *Theoretical Computer Science*, 615(C):23–44, February 2016.

15 Bogdan S. Chlebus, Leszek Gasieniec, Dariusz R. Kowalski, and Tomasz Radzik. On the wake-up problem in radio networks. In *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP)*, pages 347–359, 2005.

16 Bogdan S. Chlebus and Dariusz R. Kowalski. A better wake-up in radio networks. In *Proceedings of 23rd ACM Symposium on Principles of Distributed Computing (PODC)*, pages 266–274, 2004.

17 Marek Chrobak, Leszek Gasieniec, and Dariusz R. Kowalski. The wake-up problem in multihop radio networks. *SIAM Journal on Computing*, 36(5):1453–1471, 2007.

18 Richard Cole, Alan M. Frieze, Bruce M. Maggs, Michael Mitzenmacher, Andréa W. Richa, Ramesh K. Sitaraman, and Eli Upfal. On balls and bins with deletions. In *Proceedings of the Second International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM)*, pages 145–158, 1998.

19 A. Czumaj and V. Stemann. Randomized Allocation Processes. In *Proceedings 38th Annual Symposium on Foundations of Computer Science*, pages 194–203, 1997.

20 Gianluca De Marco and Grzegorz Stachowiak. Asynchronous shared channel. In *Proceedings of the ACM Symposium on Principles of Distributed Computing*, PODC '17, pages 391–400, 2017.

21 Devdatt Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 1st edition, 2009.

22 Devdatt Dubhashi and Desh Ranjan. Balls and Bins: A Study in Negative Dependence. *Random Structures & Algorithms*, 13(2):99–124, 1998. `doi:10.1002/(SICI)1098-2418(199809)13:2<99::AID-RSA1>3.0.CO;2-M`.

23 Jeremy T. Fineman, Seth Gilbert, Fabian Kuhn, and Calvin Newport. Contention resolution on a fading channel. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC)*, pages 155–164, 2016.

24 Jeremy T. Fineman, Calvin Newport, and Tonghe Wang. Contention resolution on multiple channels with collision detection. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing, PODC 2016, Chicago, IL, USA, July 25-28, 2016*, pages 175–184, 2016.

25 Mihály Geréb-Graus and Thanasis Tsantilas. Efficient optical communication in parallel computers. In *Proceedings 4th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 41–48, 1992.

26 Leslie Ann Goldberg and Philip D. MacKenzie. Analysis of practical backoff protocols for contention resolution with multiple servers. *Journal of Computer and System Sciences*, 58(1):232–258, 1999. `doi:10.1006/jcss.1998.1590`.

27 Leslie Ann Goldberg, Philip D. Mackenzie, Mike Paterson, and Aravind Srinivasan. Contention resolution with constant expected delay. *Journal of the ACM*, 47(6):1048–1096, 2000.

28 Jonathan Goodman, Albert G. Greenberg, Neal Madras, and Peter March. Stability of binary exponential backoff. *Journal of the ACM*, 35(3):579–602, July 1988.

29 Ronald I. Greenberg and Charles E. Leiserson. Randomized routing on fat-trees. In *Proceedings of the 26th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 241–249, 1985.

30 Johan Hastad, Tom Leighton, and Brian Rogoff. Analysis of backoff protocols for multiple access channels. *SIAM Journal on Computing*, 25(4):1996, 740-774.

31 IEEE. IEEE standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks – Specific requirements - Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pages 1–3534, 2016.

32 Tomasz Jurdzinski and Grzegorz Stachowiak. The cost of synchronizing multiple-access channels. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC)*, pages 421–430, 2015.

33 A R Karlin and E Upfal. Parallel Hashing - An Efficient Implementation of Shared Memory. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing (STOC)*, pages 160–168, 1986.

34 James F. Kurose and Keith Ross. *Computer Networking: A Top-Down Approach*. Pearson, 6th edition, 2013.

35 Christoph Lenzen and Roger Wattenhofer. Tight Bounds for Parallel Randomized Load Balancing: Extended Abstract. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 11–20, 2011.

36 Gianluca De Marco and Dariusz R. Kowalski. Fast nonadaptive deterministic algorithm for conflict resolution in a dynamic multiple-access channel. *SIAM Journal on Computing*, 44(3):868–888, 2015.

37 Gianluca De Marco and Dariusz R. Kowalski. Contention resolution in a non-synchronized multiple access channel. *Theoretical Computer Science*, 689:1–13, 2017.

38 Robert M. Metcalfe and David R. Boggs. Ethernet: Distributed packet switching for local computer networks. *Communications of the ACM*, 19(7):395–404, July 1976.

39 Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, New York, NY, USA, 2005.

40 Michael David Mitzenmacher. *The Power of Two Choices in Randomized Load Balancing*. PhD thesis, University of California, Berkeley, 1996.

41 K. Nakano and S. Olariu. Uniform leader election protocols for radio networks. *IEEE Transactions on Parallel and Distributed Systems*, 13(5):516–526, May 2002. `doi:10.1109/TPDS.2002.1003864`.

42 Adrian Ogierman, Andrea Richa, Christian Scheideler, Stefan Schmid, and Jin Zhang. Sade: competitive MAC under adversarial SINR. *Distributed Computing*, 31(3):241–254, June 2018.

43 Prabhakar Raghavan and Eli Upfal. Stochastic contention resolution with short delays, April 1999.

44 Andrea Richa, Christian Scheideler, Stefan Schmid, and Jin Zhang. A jamming-resistant MAC protocol for multi-hop wireless networks. In *Proceedings of the International Symposium on Distributed Computing (DISC)*, pages 179–193, 2010.

45 Andrea Richa, Christian Scheideler, Stefan Schmid, and Jin Zhang. Competitive and fair medium access despite reactive jamming. In *Proceedings of the* 31st *International Conference on Distributed Computing Systems (ICDCS)*, pages 507–516, 2011.

46 Andrea Richa, Christian Scheideler, Stefan Schmid, and Jin Zhang. Competitive and Fair Throughput for Co-existing Networks Under Adversarial Interference. In *Proceedings of the 2012 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 291–300, 2012.

47 Andrea W Richa, M Mitzenmacher, and R Sitaraman. The power of two random choices: A survey of techniques and results. *Combinatorial Optimization*, 9:255–304, 2001.

48 X. Sun and L. Dai. Backoff Design for IEEE 802.11 DCF Networks: Fundamental Tradeoff and Design Criterion. *IEEE/ACM Transactions on Networking*, 23(1):300–316, 2015.

49 Eli Upfal. Efficient Schemes for Parallel Communication. *J. ACM*, 31(3):507–517, June 1984.

50 Berthold Vöcking. How asymmetry helps load balancing. *Journal of the ACM*, 50(4):568–589, 2003.

51 Dan E. Willard. Log-logarithmic selection resolution protocols in a multiple access channel. *SIAM J. Comput.*, 15(2):468–477, May 1986.

52 D. Yin, K. Lee, R. Pedarsani, and K. Ramchandran. Fast and Robust Compressive Phase Retrieval with Sparse-Graph Codes. In *2015 IEEE International Symposium on Information Theory (ISIT)*, pages 2583–2587, June 2015.

Appendix

A Chernoff Bounds and Property 1

In Problem 1.8 of Dubhashi and Panconesi [21], the following question is posed: Show that if Property 1 holds, then Theorem 3 holds. We are invoking this result, but an argument is absent in [21].

We bridge this gap with Claim 15 below. This fits directly into the derivation of Chernoff bounds given in Dubhashi and Panconesi [21]. In particular, the line above Equation 1.3 on page 4 of [21] claims equality for Equation 10 below by invoking independence of the random variables. Here, Claim 15 gives an inequality (in the correct direction) and the remainder of the derivation in [21] follows without any further modifications.

▷ Claim 15. Let $X_1, \cdots, X_n$ be a set of indicator random variables satisfying the property:

$$Pr\left[\bigwedge_{i\in\$} X_i = 1\right] \leq \prod_{i\in\$} Pr\left[X_i = 1\right] \tag{9}$$

for all subsets $\$ \subset \{1, \cdots, n\}$. Then the following holds:

$$E\left[\prod_{i=1}^{n} e^{\lambda X_i}\right] \leq \prod_{i=1}^{n} E\left[e^{\lambda X_i}\right] \tag{10}$$

Proof. Let $\mathbb{N}$ denote the set of strictly positive integers. First, we need to point out two properties of indicator random variables

(i) $X_i^k = X_i$ for all $k \in \mathbb{N}$; and

(ii) $E\left[X_i\right] = Pr\left[X_i = 1\right]$, and $E\left[\prod_{i\in\$} X_i\right] = Pr\left[\bigwedge_{i\in\$} X_i = 1\right]$ for all subset $\$$.

By Taylor expansion we have $e^{\lambda X_i} = \sum_{k=0}^{\infty} \lambda^k \frac{X_i^k}{k!}$, and then,

$$E\left[e^{\lambda X_i}\right] = \sum_{k=0}^{\infty} \lambda^k \frac{E\left[X_i^k\right]}{k!} \tag{11}$$

Thus, the product in the left hand side (LHS) of (10) becomes $\prod_{i=1}^{n} e^{\lambda X_i} = \prod_{i=1}^{n} \left(\sum_{k=0}^{\infty} \frac{\lambda^k}{k!} X_i^k\right)$, which can be written as a polynomial function of λ, i.e. $\sum_{r=0}^{\infty} f_r \lambda^r$, where f_r are coefficients which may contain the indicator random variables X_is. Here $f_0 = 1$. To get the expression of f_r for $r \geq 1$, we first define a set, for all integers $k, r \in \mathbb{N}$ with $k \leq r$, let $\mathcal{I}(k, r) = \{(d_1, d_2, \cdots, d_k) : d_1, \cdots, d_k \in \mathbb{N}, d_1 \leq d_2 \leq \cdots \leq d_k, d_1 + d_2 + \cdots + d_k = r\}$. Then the coefficients f_r, $r \geq 1$, can be expressed as

$$f_r = \sum_{k=1}^{\min\{r,n\}} \sum_{(d_1,\cdots,d_k)\in\mathcal{I}(r,k)} \sum_{1\leq i_1\neq i_2\neq\cdots\neq i_k\leq n} \frac{X_{i_1}^{d_1}}{d_1!} \frac{X_{i_2}^{d_2}}{d_2!} \cdots \frac{X_{i_k}^{d_k}}{d_k!}. \tag{12}$$

For example,

$$
\begin{aligned}
f_1 &= \sum_{i=1}^{n} X_i \\
f_2 &= \sum_{i=1}^{n} \frac{X_i^2}{2!} + \sum_{1\le i_1\ne i_2\le n} X_{i_1} X_{i_2} \\
f_3 &= \sum_{i=1}^{n} \frac{X_i^3}{3!} + \sum_{1\le i_1\ne i_2\le n} X_{i_1} \frac{X_{i_2}^2}{2!} + \sum_{1\le i_1\ne i_2\ne n_3\le n} X_{i_1} X_{i_2} X_{i_3} \\
&\vdots
\end{aligned}
$$

With the expression (12), the LHS becomes

$$
\begin{aligned}
\mathrm{LHS} &= 1 + \sum_{r=1}^{\infty} \lambda^r \sum_{k=1}^{\min\{r,n\}} \sum_{(d_1,\cdots,d_k)\in\mathcal{I}(r,k)} \sum_{1\le i_1\ne i_2\ne\cdots\ne i_k\le n} E\left[\frac{X_{i_1}^{d_1}}{d_1!}\frac{X_{i_2}^{d_2}}{d_2!}\cdots\frac{X_{i_k}^{d_k}}{d_k!}\right] \\
&= 1 + \sum_{r=1}^{\infty} \lambda^r \sum_{k=1}^{\min\{r,n\}} \sum_{(d_1,\cdots,d_k)\in\mathcal{I}(r,k)} \sum_{1\le i_1\ne i_2\ne\cdots\ne i_k\le n} \frac{E\left[X_{i_1}^{d_1}X_{i_2}^{d_2}\cdots X_{i_k}^{d_k}\right]}{d_1!d_2!\cdots d_k!}
\end{aligned}
$$

Similarly, with the Taylor expansion of (11), the product in the right hand side (RHS) of (10) becomes

$$
\begin{aligned}
RHS &= \prod_{i=1}^{n}\left(\sum_{k=0}^{\infty}\lambda^k \frac{E\left[X_i^k\right]}{k!}\right) \\
&= 1 + \sum_{r=1}^{\infty} \lambda^r \sum_{k=1}^{\min\{r,n\}} \sum_{(d_1,\cdots,d_k)\in\mathcal{I}(r,k)} \sum_{1\le i_1\ne i_2\ne\cdots\ne i_k\le n} \frac{E\left[X_{i_1}^{d_1}\right]}{d_1!}\frac{E\left[X_{i_2}^{d_2}\right]}{d_2!}\cdots\frac{E\left[X_{i_k}^{d_k}\right]}{d_k!} \\
&= 1 + \sum_{r=1}^{\infty} \lambda^r \sum_{k=1}^{\min\{r,n\}} \sum_{(d_1,\cdots,d_k)\in\mathcal{I}(r,k)} \sum_{1\le i_1\ne i_2\ne\cdots\ne i_k\le n} \frac{E\left[X_{i_1}^{d_1}\right]E\left[X_{i_2}^{d_2}\right]\cdots E\left[X_{i_k}^{d_k}\right]}{d_1!d_2!\cdots d_k!}
\end{aligned}
$$

By the above-mentioned two properties (i) and (ii) of indicator random variables, then

$$
\begin{aligned}
E\left[X_{i_1}^{d_1}X_{i_2}^{d_2}\cdots X_{i_k}^{d_k}\right] &= E\left[X_{i_1}X_{i_2}\cdots X_{i_k}\right] = Pr\left[X_{i_1}=1, X_{i_2}=1,\cdots,X_{i_k}=1\right] \\
E\left[X_{i_1}^{d_1}\right]E\left[X_{i_2}^{d_2}\right]\cdots E\left[X_{i_k}^{d_k}\right] &= E\left[X_{i_1}\right]E\left[X_{i_2}\right]\cdots E\left[X_{i_k}\right] \\
&= Pr\left[X_{i_1}=1\right]Pr\left[X_{i_2}=1\right]\cdots Pr\left[X_{i_k}=1\right].
\end{aligned}
$$

By the condition (9), we have $Pr\left[X_{i_1}=1, X_{i_2}=1,\cdots,X_{i_k}=1\right] \le Pr\left[X_{i_1}=1\right]Pr\left[X_{i_2}=1\right]\cdots Pr\left[X_{i_k}=1\right]$, and thus

$$
E\left[X_{i_1}^{d_1}X_{i_2}^{d_2}\cdots X_{i_k}^{d_k}\right] \le E\left[X_{i_1}^{d_1}\right]E\left[X_{i_2}^{d_2}\right]\cdots E\left[X_{i_k}^{d_k}\right].
$$

Thus (10) holds. ◁