

# Instability of backoff protocols with arbitrary arrival rates

Leslie Ann Goldberg and John Lapinskas\*

## Abstract

In contention resolution, multiple processors are trying to coordinate to send discrete messages through a shared channel with sharply limited communication. If two processors inadvertently send at the same time, the messages collide and are not transmitted successfully. An important case is acknowledgement-based contention resolution, in which processors cannot listen to the channel at all; all they know is whether or not their own messages have got through. This situation arises frequently in both networking and cloud computing. The most common acknowledgement-based protocols in practice are backoff protocols — variants of binary exponential backoff are used in both Ethernet and TCP/IP, and both Google Drive and AWS instruct their users to implement it to handle busy periods.

In queueing models, where each processor has a queue of messages, stable backoff protocols are already known (Håstad et al., SICOMP 1996). In queue-free models, where each processor has a single message but processors arrive randomly, it is a long-standing conjecture of Aldous (IEEE Trans. Inf. Theory 1987) that no stable backoff protocols exist for any positive arrival rate of processors. Despite exciting recent results for full-sensing protocols which assume far greater listening capabilities of the processors (see e.g. Bender et al. STOC 2020 or Chen et al. PODC 2021), this foundational question remains open; here instability is only known in general when the arrival rate of processors is at least 0.42 (Goldberg et al. SICOMP 2004). We prove Aldous’s conjecture for all backoff protocols outside of a tightly-constrained special case using a new domination technique to get around the main difficulty, which is the strong dependencies between messages.

## 1 Introduction

In the field of contention resolution, multiple processors (sometimes called “stations”) are trying to coordinate to send discrete messages (sometimes called “packets”) through a shared channel called a multiple access channel. The multiple access channel is not centrally controlled and the processors cannot communicate, except by listening to the channel. The operation of the channel is straightforward. In each (discrete) time step one or more processors might send messages to the channel. If exactly one message is sent then it is delivered successfully and the sender is notified of the success. If multiple messages are sent then they collide and are not transmitted successfully (so they will have to be re-sent later).

We typically view the entire process as a discrete-time Markov chain. At each time step, new messages arrive at processors with rates governed by probability distributions with total rate  $\lambda > 0$ . After the arrivals, each processor independently chooses whether to send a message through the channel. A *contention-resolution protocol* is a randomised algorithm that the processors use to decide when to send messages to the channel (and when to wait because the channel is too busy!). Our objective is to find a *stable* protocol [15], which is a protocol with the property that the

---

\*For the purpose of Open Access, the authors have applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. All data is provided in full in the results section of this paper.

corresponding Markov chain is positive recurrent, implying that there is a stationary distribution bounding the likely extent to which messages build up over time. Other objectives include bounding the expected waiting time of messages and maximising the throughput, which is the rate at which messages succeed.

Issues of contention resolution naturally arise when designing networking protocols [1, 2], but it is also relevant to hardware design that enables concurrency [18, 23] and to interaction with cloud computing services [9, 10]. The example of cloud computing will be an instructive one, so we expand on it. Suppose that an unknown number of users is submitting requests to a server which is struggling under the load, with more users arriving over time. The users do not have access to load information from the server and they do not have knowledge of each other — all they know is whether or not their own requests to the server are getting through. The central question of contention resolution is then: how often should users re-send their requests in order to get everyone’s requests through as quickly as possible?

There are two main categories of contention resolution protocol — these differ according to the extent to which processors listen to the channel. In *full-sensing protocols*, processors constantly listen to the shared channel obtaining partial information. For example, in addition to learning whether its own sends are successful, a processor may learn on which steps the channel is quiet (with no sends) [26] or it may learn on which steps there are successful sends [6] or it may learn both [21]. While full-sensing protocols are suitable in many settings, there are important settings where they cannot reasonably be implemented (such as the cloud computing example above). In *acknowledgement-based protocols*, the only information that processors receive about the shared channel is whether their own messages get through.

We distinguish between two ways of modelling message arrival. The earliest work in the field focused on *queueing models*, in which the number  $N$  of processors is fixed and each processor maintains a queue of messages to be sent. These models are appropriate for a static network. A particularly simple example is the slotted ALOHA protocol [24], one of the first networking protocols. In this protocol, if there are  $N$  processors with non-empty queues, then these processors send independently with probability  $1/N$ . For large  $N$ , this is stable if  $\lambda < 1/e$ . However, it requires the processors to know the value of  $N$ . In order to get around this difficulty, Metcalfe and Boggs proposed *binary exponential backoff*, in which a processor which has already had  $i$  unsuccessful attempts to send a given message waits a random amount of time (a geometric random variable with mean  $2^i$ ) before again attempting to send. Binary exponential backoff (with some modifications) forms the basis for Ethernet [20] and TCP/IP [5]. For any  $N$ , binary exponential backoff is known to be stable for sufficiently small  $\lambda$  [15, 3]. Unfortunately, this value of  $\lambda$  depends on  $N$  and binary exponential backoff is unstable if  $\lambda$  is sufficiently large [17]. Remarkably, Håstad, Leighton and Rogoff [17] showed that *polynomial backoff* (where the waiting time after the  $i$ ’th collision is a geometric random variable with expectation  $i^\alpha$  for some  $\alpha > 1$ ) is stable for all  $\lambda \in (0, 1)$ . For contention resolution with queues even more powerful full-sensing protocols are known — in particular, there is a stable full-sensing protocol even for the more general model in which some specified pairs of processors are allowed to use the channel simultaneously [25, 26].

In this paper, we focus on *queue-free models*, which allow for dynamic networks and are more appropriate for public wi-fi hotspots [1] or cloud computing [9, 10]. We again consider these models in discrete time. In these models, processors arrive in the system according to a Poisson distribution with rate  $\lambda$ , and each processor only wants to send a single message rather than maintaining a queue; in fact, we typically identify the processors with the messages that they are trying to send. As usual, only one message can pass through the channel at any given time step. In this setting, an acknowledgement-based protocol can be viewed as a joint distribution  $(T_1, T_2, \dots)$  of times. For each message, the corresponding processor independently samples  $(\tau_1, \tau_2, \dots)$  from  $(T_1, T_2, \dots)$ . If

the message does not get through during the first  $j - 1$  times that it is sent then the processor waits for  $\tau_j$  time steps before sending it for the  $j$ 'th time. An important special case is that of *backoff protocols*, in which  $(T_1, T_2, \dots)$  is a tuple of independent geometric variables. Equivalently, a backoff protocol is associated with a *send sequence* of probabilities  $\mathbf{p} = p_0, p_1, \dots$  such that, if a processor has already had  $j$  unsuccessful sends, then it will send its message on the following time step with probability  $p_j$ ; thus  $1/p_j$  is the expected waiting time  $\mathbb{E}(T_j)$ . For example, the case  $p_j = 2^{-j}$  gives rise to the binary exponential backoff protocol that we have already described. This protocol is widely used in the queue-free model: both AWS and Google Drive advise users to implement binary exponential backoff when using their services [9, 10].

In the queue-free setting, there has been a great deal of interesting work developing full-sensing protocols and proving that these perform well. Some of this is described in the survey of Chlebus [8]. See also [21, 11, 16]. More recently, Bender et al. [6] have shown that in the full-sensing model without collision detection (where processors listen to the channel to learn on which steps there are successful sends but are unable to distinguish between silence and collisions) there is a full-sensing protocol which achieves constant throughput even when the message arrival is adversarial rather than random. Chen et al. [7] demonstrate a full-sensing protocol that can achieve a decent throughput, even in the presence of jamming. Despite these advances regarding full-sensing protocols, and other protocols assuming more capabilities from processors than acknowledgement-based protocols [22, 14], for acknowledgement-based protocols the most fundamental possible question remains open: **do stable protocols exist at all?** Indeed, this problem remains open even for backoff protocols, and most work on the question has focused on this case.

The following foundational conjecture was made by Aldous [4] in 1987, and is widely believed. It is the focus of this work.

**Conjecture 1** (Aldous's Conjecture). *In the queue-free setting, no backoff protocol is stable for any positive value of  $\lambda$ .*

Aldous's conjecture remains open to this day. It has been proved for arrival rates  $\lambda \geq 0.42$  [12], but for arbitrary arrival rates the only known results concern special cases which avoid a central difficulty inherent to the problem. Consider a backoff protocol with send sequence  $p_0, p_1, \dots$  and arrival rate  $\lambda > 0$ . For all integers  $j, t \geq 1$ , write  $b_j(t)$  for the set of messages in the system at time  $t$  which have already sent  $j$  times (all unsuccessfully). Write  $S_j(t)$  for the number of messages in  $b_j(t)$  which send at time  $t$ ,  $S_0(t)$  for the number of "newborn" messages which send for the first time at time  $t$ , and  $S(t) = S_0(t) + S_1(t) + \dots$  for the total number of sends from all messages in the system. Thus a message escapes the system at time  $t$  if and only if  $S(t) = 1$ , Aldous's conjecture implies that  $S(t) = 1$  for less than a  $\lambda$  proportion of times, i.e. that messages arrive faster than they escape. Very often, the reason that this occurs is that  $S(t) \geq 2$  for most values of  $t$ . However, it is not hard to show that for all  $j \geq 1$  and most times  $t$  we have  $\mathbb{E}(S_j(t)) = p_j \mathbb{E}(|b_j(t)|) \lesssim \lambda$ , and so on most time steps  $S_j(t) = 0$ ; thus to show  $S(t) \geq 2$  on most time steps, we must engage with the complicated joint distribution  $(|b_1(t)|, |b_2(t)|, \dots)$ . This is the key difficulty that all current arguments have avoided, which restricts the classes of send sequences to which they apply. The tool that enabled us to prove Aldous's conjecture for most protocols is a new domination technique for bounding this joint distribution. Before stating our result and the new technique we first summarise progress that can be made without engaging with the joint distribution. In the following summary, we classify protocols in terms of the key quantity  $1/p_j$ , which is the expected waiting time before a message sends after having its  $j$ 'th collision.

- Kelly and MacPhee [19] categorised the class of backoff protocols for which  $S(t) \geq 2$  for all sufficiently large  $t$ . This result covers all protocols with subexponential expected waiting

times, i.e. whose send sequences satisfy  $1/p_j = o(c^j)$  as  $j \rightarrow \infty$  for all  $c > 1$  (see Corollary 9). Since these protocols are unstable in such a very strong way, Kelly and MacPhee are able to avoid working with the joint distribution in favour of applying the Borel-Cantelli lemmas.

- Aldous [4] proved that binary exponential backoff is unstable for all  $\lambda > 0$ , and his argument easily extends to all backoff protocols with exponential expected waiting times, i.e. whose send sequences satisfy  $1/p_j = \Theta(c^j)$  as  $j \rightarrow \infty$  for some  $c > 1$  (see Section 3). This proof relies on proving concentration for specific variables  $|b_j(t)|$  as  $t \rightarrow \infty$ , and then applying union bounds over suitable ranges of  $j$  and  $t$ , again avoiding the joint distribution. This concentration fails in general; for example, if  $p_j \geq 3\lambda$ , then  $b_j(t) = \emptyset$  for most values of  $t$ . (See Definition 10 for a more detailed discussion.)
- A simple argument known to the authors of [12] (but not previously published) shows that there is no stable backoff protocol which has infinitely many super-exponential expected waiting times, i.e. with an infinite subsequence  $p_{j_1}, p_{j_2}, \dots$  satisfying  $1/p_{j_k} = \omega(c^{j_k})$  as  $k \rightarrow \infty$  for all  $c > 1$ . We state and prove this as Lemma 7. In this case there is no need to engage with the joint distribution of the  $S_j(t)$  variables because the proof relies on bounding  $S(t) \geq S_0(t)$ , i.e. only considering newborn messages.

Unfortunately, the above results cannot be combined in any simple way to prove Aldous's conjecture. For example, by including some  $p_j$  values such that the expected waiting time  $1/p_j$  is less than exponential, it is easy to construct protocols which neither exhibit concentration for specific variables  $|b_j(t)|$  nor satisfy  $S(t) \geq 2$  for all sufficiently large  $t$  (see Section 3), so to show that these protocols are unstable we must engage with the joint distribution.

Our main technical contribution is a proof (see Lemma 14) that, roughly speaking, we can dominate the joint distribution of  $(|b_1(t)|, |b_2(t)|, \dots)$  below by a much simpler collection of independent Poisson variables whenever  $\mathbb{E}(S(t)) \rightarrow \infty$  as  $t \rightarrow \infty$ . Using this, we are able to almost entirely solve the problem of inconsistent decay rates and prove Aldous's conjecture except in some extreme cases characterised in Definition 4. Before describing these extreme cases, we give some easier-to-state consequences of our main result (Theorem 5). In the following theorems (and throughout the paper) a *backoff process* is a backoff protocol in the queue-free model. The first consequence is that all protocols with monotonically non-increasing send sequences are unstable.

**Theorem 2.** *For every  $\lambda \in (0, 1)$  and every monotonically non-increasing send sequence  $\mathbf{p} = p_0, p_1, \dots$ , the backoff process with arrival rate  $\lambda$  and send sequence  $\mathbf{p}$  is unstable.*

We have included Theorem 2 because it has a clean statement, but our proof technique doesn't rely on any kind of monotonicity. For example, our main Theorem, Theorem 5 also has the following corollary.

**Theorem 3.** *Let  $\mathbf{p}$  be a send sequence. Let  $m_{\mathbf{p}}(n)$  be the median of  $p_0, \dots, p_n$ . Suppose that  $m_{\mathbf{p}}(n) = o(1)$ . Then for every  $\lambda \in (0, 1)$  the backoff process with arrival rate  $\lambda$  and send sequence  $\mathbf{p}$  is unstable.*

Of course, there is nothing very special about the median. The same would be true of sequences for which any centile is  $o(1)$ . At this point, we are ready to describe the extreme cases that elude our new proof technique, and to state our main result, which shows instability except in these extreme cases. The extreme cases have the property that the send sequence is almost entirely constant, with occasional exponential (but not super-exponential) waiting times thrown in.

**Definition 4.** *A send sequence  $\mathbf{p}$  is LCED ("largely constant with exponential decay") if it satisfies the following properties:*

- (i) **“Largely constant”**: For all  $\eta > 0$ , there exists  $c > 0$  such that for infinitely many  $n$ ,  $|\{j \leq n: p_j > c\}| \geq (1 - \eta)n$ .
- (ii) **“with exponential decay”**:  $\mathbf{p}$  has an infinite subsequence  $(p_{\ell_1}, p_{\ell_2}, \dots)$  which satisfies  $\log(1/p_{\ell_x}) = \Theta(\ell_x)$  as  $x \rightarrow \infty$ .
- (iii) **“(but without super-exponential decay)”**:  $\log(1/p_j) = O(j)$  as  $j \rightarrow \infty$ .

As an illustrative example of item (i) taking  $\eta = 999/1000$ , it implies that as you progress along the send sequence  $p_1, p_2, \dots$ , infinitely often, you will notice that all but  $1 - \eta = 0.1\%$  of the  $p_j$ ’s you have seen so far are bounded below by some constant  $c$ . The same holds for values of  $\eta$  that are closer to 1, but  $c$  will be correspondingly smaller. Obviously, this is also true for values of  $\eta$  that are closer to 1 but the corresponding constant  $c$  would be smaller. Item (ii) means that there is an infinite subsequence of  $j$ ’s where the expected waiting times (after  $j$  failures) is exponentially long. Item (iii) just means that expected waiting times are not more than exponentially long.

With this definition, we can state our main theorem, which proves Aldous’s conjecture for all sequences except LCED sequences and extends all previously-known results.

**Theorem 5.** *Let  $\mathbf{p}$  be a send sequence which is not LCED. Then for every  $\lambda \in (0, 1)$  the backoff process with arrival rate  $\lambda$  and send sequence  $\mathbf{p}$  is unstable.*

As we discuss in Section 5, LCED sequences can exhibit qualitatively different behaviour from non-LCED sequences, with arbitrarily long “quiet patches” during which almost every message that sends is successful. Lemma 14, our domination of the joint distribution  $(|b_1(t)|, |b_2(t)|, \dots)$  by independent Poisson variables, is actually false during these “quiet patches”, and so proving Aldous’s conjecture for LCED sequences will require new ideas. On a conceptual level, the “quiet patches” exhibited by some LCED sequences are essentially the only remaining obstacle to a full proof of Aldous’s conjecture.

Nevertheless, as we show in this paper, our new domination is sufficient to cover all send sequences except the LCED sequences. Thus Lemma 14 makes substantial progress on the long-standing conjecture, the first progress in many years, resulting in strong new instability results such as Theorems 2 and 3.

The remainder of the paper is structured as follows. Section 2 gives the formal definition of a backoff process. Sections 3 and 4 sketch the proof of Theorem 5, with Section 3 giving an overview of the relevant existing proof techniques and Section 4 explaining our novel ideas. Section 5 discusses the remaining obstacles to proving Aldous’s conjecture. In this paper we present only sketch proofs of our results, but a full version with proofs is available at [13]

## 2 Formal definitions

We say that a stochastic process is *stable* if it is positive recurrent, and *unstable* otherwise (i.e. if it is null recurrent or transient). A *backoff process* is a backoff protocol in the queue-free model.

Informally, a backoff process is a discrete-time Markov chain associated with an arrival rate  $\lambda \in (0, 1)$  and a send sequence  $\mathbf{p} = p_0, p_1, p_2, \dots$  of real numbers in the range  $(0, 1]$ . Following Aldous [4], we identify processors and messages, and we think of these as balls moving through a sequence of bins. Each time a message sends, if no other message sends at the same time step, it leaves the system; otherwise, it moves to the next bin. Thus at time  $t$ , the  $j$ ’th bin contains all messages which have sent  $j$  times without getting through (these sends occurred at time steps up to and including time  $t$ ). The system then evolves as follows at a time step  $t$ . First, new messages are added to bin 0 according to a Poisson distribution with rate  $\lambda$ . Second, for all  $j \geq 0$ , each message in bin  $j$  sends independently with probability  $p_j$ . Third, if exactly one message sends then

it leaves the system, and otherwise all messages that sent from any bin  $j$  move to the next bin, bin  $j + 1$ .

**Remark 6.** *There is no need to consider arrival rates  $\lambda \geq 1$  because it is already known that backoff processes with arrival rate  $\lambda \geq 1$  are unstable [12]. We also don't allow  $p_j = 0$  since that would trivially cause transience (hence, instability).*

**Formal definition of backoff processes.** A backoff process with arrival rate  $\lambda \in (0, 1)$  and send sequence  $\mathbf{p} = p_0, p_1, p_2, \dots \in (0, 1]$  is a stochastic process  $X$  defined as follows. Time steps  $t$  are positive integers. Bins  $j$  are non-negative integers. There is an infinite set of balls. We now define the set  $b_j^X(t)$ , which will be the set of balls in bin  $j$  just after (all parts of) the  $t$ 'th step. Initially, all bins are empty, so for all non-negative integers  $j$ ,  $b_j^X(0) = \emptyset$ . For any positive integer  $t$ , the  $t$ 'th step of  $X$  involves (i) step initialisation (including birth), (ii) sending, and (iii) adjusting the bins. Step  $t$  proceeds as follows.

- Part (i) of step  $t$  (step initialisation, including birth): An integer  $n_t$  is chosen independently from a Poisson distribution with mean  $\lambda$ . This is the number of *newborns* at time  $t$ . The set  $b_0^X(t)$  contains the balls in  $b_0^X(t-1)$  together with  $n_t$  new balls which are *born* at time  $t$ . For each  $j \geq 1$  we define  $b_j^X(t) = b_j^X(t-1)$ .
- Part (ii) of step  $t$  (sending): For all  $j \geq 0$ , all balls in  $b_j^X(t)$  send independently with probability  $p_j$ . We use  $\mathbf{send}^X(t)$  for the set of balls that send at time  $t$ .
- Part (iii) of step  $t$  (adjusting the bins):
  - If  $|\mathbf{send}^X(t)| \leq 1$  then any ball in  $\mathbf{send}^X(t)$  *escapes* so for all  $j \geq 0$  we define  $b_j^X(t) = b_j^X(t) \setminus \mathbf{send}^X(t)$ .
  - Otherwise, no balls escape but balls that send move to the next bin, so we define  $b_0^X(t) = b_0^X(t) \setminus \mathbf{send}^X(t)$  and, for all  $j \geq 1$ ,  $b_j^X(t) = (b_{j-1}^X(t) \cap \mathbf{send}^X(t)) \cup (b_j^X(t) \setminus \mathbf{send}^X(t))$ .

Finally, we define  $\mathbf{balls}^X(t) = \cup_j b_j^X(t)$ .

### 3 Technical context

We first formally state the result alluded to in Section 1 which proves instability for backoff protocols whose send sequences decay super-exponentially.

**Lemma 7.** *Let  $X$  be a backoff process with arrival rate  $\lambda \in (0, 1)$  and send sequence  $\mathbf{p} = p_0, p_1, \dots$ . If, for infinitely many  $j$ ,  $p_j \leq (\lambda p_0/2)^j$ , then  $X$  is unstable.*

We defer the proof to Section 2 of the full version, but it is simple. Essentially, we dominate the expected time for a newborn ball to leave the process below under the assumption that a sending ball always leaves the process unless another ball is born at the same time. Under the assumptions of Lemma 7, this is infinite.

We next describe the results of Kelly and MacPhee [19] and Aldous [4] in more detail than the previous section. This will allow us in Section 4 to clearly identify the regimes in which instability is not known, and to clearly highlight the novel parts of our arguments.

We first introduce a key notion from Aldous [4]. Informally, an *externally-jammed process* is a backoff process in which balls never leave; thus if a single ball sends at a given time step, it moves to the next bin as normal. (See Section 3.1 of the full version for a formal definition.) Unlike backoff processes, an externally-jammed process starts in its stationary distribution; thus for all  $j \geq 0$ ,



the size of  $b_j(0)$  is drawn from a Poisson distribution with mean  $\lambda/p_j$ . There is a natural coupling between a backoff process  $X$  and an externally-jammed process  $Y$  such that  $|b_j^X(t)| \leq |b_j^Y(t)|$  for all  $j$  and  $t$  (see Observation 22 of the full version); thus an externally-jammed process can be used to dominate (from above) the number of balls in a backoff process.

As discussed earlier, Kelly and MacPhee [19] gives a necessary and sufficient condition for infinitely many messages to get through. In our context, the relevant case of their result can be stated as follows. Given a send sequence  $\mathbf{p}$ , let  $W_0, W_1, \dots$  be independent geometric variables such that  $W_j$  has parameter  $p_j$  for all  $j$ . Then for all  $\tau \geq 0$ , define  $\mu_\tau(\mathbf{p}) = \sum_{j=0}^{\infty} \mathbb{P}\left(\sum_{k=0}^j W_k \leq \tau\right)$ . If a ball is born at time  $t$  in an externally-jammed process,  $\mu_\tau(\mathbf{p})$  is the expected number of times that ball sends up to time  $t + \tau$ .

**Theorem 8** ([19, Theorem 3.10]). *Let  $\mathbf{p}$  be a send sequence, and suppose that for all  $\lambda \in (0, 1)$ , we have  $\sum_{\tau=0}^{\infty} \mu_\tau(\mathbf{p}) e^{-\lambda \mu_\tau(\mathbf{p})} < \infty$ . Then for all  $\lambda \in (0, 1)$ , the backoff process  $X$  with arrival rate  $\lambda$  and send sequence  $\mathbf{p}$  is unstable. Moreover, with probability 1, only finitely many balls leave  $X$ .*

The following corollary is proved in Section 2 of the full version.

**Corollary 9.** *Let  $\mathbf{p}$  be a send sequence such that  $\log(1/p_j) = o(j)$  as  $j \rightarrow \infty$ . Then for all  $\lambda \in (0, 1)$ , the backoff process  $X$  with arrival rate  $\lambda$  and send sequence  $\mathbf{p}$  is unstable.*

The result that is proved in Aldous [4] is the instability of binary exponential backoff, in which  $p_j = 2^{-j}$  for all  $j$ , for all arrival rates  $\lambda > 0$ . Aldous’s paper says “without checking the details, [he] believe[s] the argument could be modified to show instability for [all backoff protocols]”. Unfortunately, this turns out not to be accurate. However, the proof does generalise in a natural way to cover a broader class of backoff protocols. Our own result extends this much further building on new ideas, but it is nevertheless based on a similar underlying framework, so we now discuss Aldous’s proof in more detail.

A key notion is that of *potential*. Informally, given a state  $\mathbf{x}(t)$  of a backoff process at time  $t$ , the potential (Definition 24 in the full version) of  $\mathbf{x}(t)$  is given by  $f(\mathbf{x}(t)) = \lambda p_0 + \sum_{j=0}^{\infty} p_j |b_j^X(t)|$ .

Thus, the potential of a backoff process at time  $t$  is the expected number of sends at time  $t + 1$  conditioned on the state at time  $t$ ; we can think of potential as “noise”. Unsurprisingly, while the potential is large, multiple balls are very likely to send at each time step so balls leave the process very slowly (see Lemma 44 of the full version).

A slightly generalised version of Aldous’s proof works as follows. The first step is to choose  $j_0$  to be a suitably large integer, and wait until a time  $t_0$  at which bins  $1, \dots, j_0$  are all “full” in the sense that  $|b_j(t_0)| \geq c\lambda/p_j$  for some constant  $c$  and all  $j \leq j_0$ . (Such a time  $t_0$  exists with probability 1. Observe that in the stationary distribution of an externally-jammed process, bin  $j$  contains  $\lambda/p_j$  balls in expectation.) We then define  $\tau_1, \tau_2, \dots$  with  $\tau_\ell = C \sum_{j=0}^{j_0+\ell} (1/p_j)$  for some constant  $C$ ; thus  $\tau_\ell$  is  $C$  times the expected number of time steps required for a newborn ball to reach bin  $b_{j_0+\ell}$  in a jammed channel. The key step of the proof is then to argue that with suitably low failure probability, for all  $\ell$  and all  $t$  satisfying  $t_0 + \tau_\ell \leq t \leq t_0 + \tau_{\ell+1}$ , all bins  $j$  with  $(j_0 + \ell)/10 \leq j \leq j_0 + \ell$  satisfy  $|b_j(t)| \geq \zeta\lambda/p_j$  for some constant  $\zeta$ . In other words, we prove that with suitably low failure probability, there is a slowly-advancing frontier of bins which are always full; in particular the potential increases to infinity, and the process is transient and hence unstable.

In order to accomplish this key step of the proof, the argument is split into two cases. If  $t$  is close to  $t_0$ , there is a simple argument based on the idea that bins 1 through  $j_0$  were full at time  $t_0$  and, since  $j_0$  is large, they have not yet had time to fully empty. For the second case, suppose that  $t$  is significantly larger than  $t_0$ , and the goal is (for example) to show that bin  $j$  is very likely to be full at time  $t = t_0 + \tau_\ell$ . By structuring the events carefully, one may assume that the process still

has large potential up to time  $t - 1$ , so with high probability not many balls leave the process in time steps  $\{t_0 + 1, \dots, t\}$ . At this point, Aldous uses a time-reversal argument to show that, in the externally-jammed process, bin  $j$  fills with balls that are born after time  $t_0$  during an interval of  $\tau_\ell$  time steps. Under a natural coupling, these balls follow the same trajectory in both the backoff process and the externally-jammed process unless they leave the backoff process; thus, by a union bound, with high probability most of these balls are present in bin  $j$  at time  $t$  as required. Aldous then applies a union bound over all bins  $j$  with  $(j_0 + \ell)/10 \leq j \leq j_0 + \ell$ ; crucially, the probability bounds for each individual bin  $j$  are strong enough that he does not need to engage with the more complicated joint distribution of the contents of the bins.

Since Aldous works only with binary exponential backoff in [4], he takes  $\tau_\ell \sim 2^{j_0 + \ell}$ ; his conjecture that his proof can be generalised to all backoff protocols is based on the idea that the definition of  $\tau_\ell$  could be modified, which gives rise to the more general argument above. Unfortunately, there are very broad classes of backoff protocols to which this generalisation cannot apply. We now define the collection of send sequences which this modified version of Aldous's proof could plausibly handle.

**Definition 10.** *A send sequence  $\mathbf{p}$  is reliable if it has the following property. Let  $\lambda > 0$ , and let  $X$  be a backoff process with arrival rate  $\lambda$  and send sequence  $\mathbf{p}$ . Then with positive probability, there exists  $\zeta > 0$ , and times  $t_0, t_1, t_2, \dots$  and collections  $\mathcal{B}_0, \mathcal{B}_1, \mathcal{B}_2, \dots$  of bins such that:*

- *for all  $i$ , for all  $t$  satisfying  $t_i \leq t \leq t_{i+1}$ , for all  $j \in \mathcal{B}_i$ , we have  $|b_j(t)| \geq \zeta \lambda / p_i$ .*
- *$|\mathcal{B}_i|$  is an increasing sequence with  $|\mathcal{B}_i| \rightarrow \infty$  as  $i \rightarrow \infty$ .*

Indeed, if Aldous's proof works for a send sequence  $\mathbf{p}$ , then it demonstrates that  $\mathbf{p}$  is reliable: Aldous takes  $\zeta = 1/2$ ,  $t_i = \tau_i$  for all  $i$ , and  $\mathcal{B}_i = \{(j_0 + i)/10, \dots, j_0 + i\}$  for all  $i$ . In short, a reliable protocol is one in which, after some “startup time”  $t_0$ , at all times one can point to a large collection of bins which will reliably *all* be full enough to provide significant potential. As discussed above, it is important that they are all full — otherwise it is necessary to delve into the complicated inter-dependencies of the bins.

## 4 Proof sketch

In this section we describe the proof of those cases of Theorem 5 not already covered by Lemma 7 and Theorem 8, encapsulated by Theorem 13 (stated below). We first set out a guiding example by defining the following family of send sequences, which we will later show is not covered by Aldous's proof techniques.

**Example 11.** *Given  $\rho \in (0, 1)$ , an increasing sequence  $\mathbf{a}$  of non-negative integers with  $a_0 = 0$ , and a function  $g: \mathbb{N} \rightarrow \mathbb{N}$ , the  $(\rho, \mathbf{a}, g)$ -interleaved send sequence  $\mathbf{p} = p_0, p_1, \dots$  is given by*

$$p_j = \begin{cases} \rho^j & \text{if } a_{2k} \leq j \leq a_{2k+1} - 1 \text{ for some } k \geq 0, \\ g(j) & \text{if } a_{2k+1} \leq j \leq a_{2k+2} - 1 \text{ for some } k \geq 0. \end{cases}$$

*Thus the  $(\rho, \mathbf{a}, g)$ -interleaved send sequence is an exponentially-decaying send sequence with base  $\rho$  spliced together with a second send sequence specified by  $g$ , with the splices occurring at points given by  $\mathbf{a}$ .*

*In this section, we will take  $g(j) = 1/\log \log j$  and  $a_k = 2^{2^k}$ . We will refer to the  $(\rho, \mathbf{a}, g)$ -interleaved send sequence with this choice of  $g$  and  $\mathbf{a}$  as a  $\rho$ -interleaved send sequence.*

Observe that a  $(\rho, \mathbf{a}, g)$ -interleaved send sequence fails to be LCED whenever  $g(j) = o(1)$ . Thus, a  $\rho$ -interleaved send sequence is not LCED.



We claim that as long as  $\rho$  is sufficiently small, a  $\rho$ -interleaved send sequence is not reliable and neither Theorem 8 nor Lemma 7 prove instability; we expand on this claim in Section 5. Our Theorem 13 will be strong enough to show that backoff protocols with these send sequences are unstable. In order to introduce Theorem 13, we first give a definition.

**Definition 12.** Fix  $\lambda, \eta$  and  $\nu$  in  $(0, 1)$ . Let

$$p_*(\lambda, \eta, \nu) = \min \left\{ \frac{\lambda}{200}, \frac{\lambda\eta}{1800 \lceil 3/\eta \rceil^2 \log(1/\nu)} \right\}.$$

A send sequence  $\mathbf{p} = p_0, p_1, \dots$  is  $(\lambda, \eta, \nu)$ -suitable if  $p_0 = 1$  and there exists  $n_0$  such that for all  $n \geq n_0$ ,

- $|\{j \in [n] \mid p_j \leq p_*(\lambda, \eta, \nu)\}| > \eta n$ , and
- $\nu^n < p_n$ .

**Theorem 13.** Fix  $\lambda, \eta$  and  $\nu$  in  $(0, 1)$ . Let  $\mathbf{p}$  be a  $(\lambda, \eta, \nu)$ -suitable send sequence. Let  $X$  be a backoff process with arrival rate  $\lambda$  and send sequence  $\mathbf{p}$ . Then  $X$  is transient, and hence unstable.

Theorem 5 will follow from Theorem 13 (actually, from a slightly more technical version of it, stated in the full version as Corollary 26), from Corollary 9, and from Lemma 7. We give the proof of Theorem 5, along with the proofs of Theorems 2 and 3, in Section 6 of the full version. Observe that for all  $\lambda, \eta > 0$ , every  $\rho$ -interleaved send sequence is a  $(\lambda, \eta, 2\rho)$ -suitable send sequence, so Theorem 13 does indeed apply.

Fix  $\lambda > 0$ . Before sketching the proof of Theorem 13, we set out one more piece of terminology. Recall that there is a natural coupling between a backoff process  $X$  and an externally-jammed process  $Y$  with arrival rate  $\lambda$  and send sequence  $\mathbf{p}$ . Under this coupling, the evolution of  $X$  is a deterministic function of the evolution of  $Y$  — balls follow the same trajectories in each process, except that they leave  $X$  if they send at a time step in which no other ball sends. As such, we work exclusively with  $Y$ , calling a ball *stuck* if it is present in  $X$  and *unstuck* if it is not. We write  $\mathbf{stuck}_j^Y(t)$  for the set of stuck balls in bin  $j$  at time  $t$ , and likewise  $\mathbf{unstuck}_j^Y(t)$  for the set of unstuck balls. (See the formal definition of an externally-jammed process in Section 3.1 of the full version.)

Intuitively, the main obstacle in applying Aldous's proof sketch to a  $\rho$ -interleaved send sequence  $\mathbf{p}$  is the long sequences of bins with  $p_j = 1/(\log \log j)$ . In the externally-jammed process  $Y$ , each individual bin  $j$  is likely to empty of balls — and hence also of stuck balls — roughly once every  $(\log j)^\lambda$  steps. (Indeed, in the stationary distribution, bin  $j$  is empty with probability  $e^{-\lambda/g(j)} = (\log j)^{-\lambda}$ .) As such, the individual bins do not provide a consistent source of potential as reliability would require. We must instead engage with the joint distribution of bins and argue that a long sequence of bins with large  $p_j$ , taken together, is likely to continue providing potential for a very long time even if individual bins empty. Unfortunately, these bins are far from independent: for example, if  $|\mathbf{stuck}_j^Y(t)| = |\mathbf{stuck}_{j+1}^Y(t)| = 0$ , then we should also expect  $|\mathbf{stuck}_{j+2}^Y(t)| = 0$ . By working only with individual reliably-full bins, Aldous was able to sidestep this issue with a simple union bound, but this is not an option for a  $\rho$ -interleaved send sequence  $\mathbf{p}$ .

Observe that since  $Y$  starts in its stationary distribution, for all  $t \geq 0$  the variables  $\{|b_j^Y(t)| : j \geq 0\}$  are mutually independent Poisson variables with  $\mathbb{E}(|b_j^Y(t)|) = \lambda/p_j$ . Intuitively, we might hope that while the potential of  $Y$  is high and very few balls are becoming unstuck, most balls in  $Y$  will remain stuck from birth and we will have  $|\mathbf{stuck}_j^Y(t)| \approx |b_j^Y(t)|$ . If this is true, we can hope to dominate the variables  $|\mathbf{stuck}_j^Y(t)|$  below by mutually independent Poisson variables as long as the potential remains large. Our largest technical contribution — and the hardest part of the proof —

is making this idea rigorous, which we do in Lemma 14. (Strictly speaking, we use a slightly more technical version stated as Lemma 45 in the full version.) This lemma is the heart of our proof; we spend Section 3 of the full version setting out the definitions needed to state it formally, then give the actual proof in Section 4 of the full version. With Lemma 14 in hand, we then prove Theorem 13 in Section 5 of the full version, using a version of Aldous’s framework in which a union bound over large bins is replaced by Chernoff bounds over a collection of independent Poisson variables which dominate a non-independent collection of smaller bins from below.

For the rest of this section, we will focus on the statement and proof of Lemma 14. A fundamental difficulty is that we can only hope for such a domination to work while the potential of  $Y$  is large, but the potential is just a weighted sum of terms  $|\text{stuck}_j^Y(t)|$ , which are precisely the variables we are concerned with — we therefore expect a huge amount of dependency. To resolve this, we define the *two-stream process* in Section 3.2 of the full version. Essentially, we split  $Y$  into a pair  $T = (Y^A, Y^B)$  of two externally-jammed processes  $Y^A$  and  $Y^B$ , each with arrival rate  $\lambda/2$ , in the natural way. We then say that a ball becomes unstuck in  $Y^A$  if it sends on a time step at which no other ball in  $Y^B$  sends, and likewise for  $Y^B$ . Thus only balls from  $Y^B$  can prevent balls in  $Y^A$  from becoming unstuck, and only balls from  $Y^A$  can prevent balls in  $Y^B$  from becoming unstuck. Each stream of the process acts as a relatively independent source of “noise” for the other stream while its potential is high. Naturally, there is a simple coupling back to the originally externally-jammed process under which stuck balls in  $Y$  are dominated below by stuck balls in  $T$ .

We must also explain what we mean by  $T$  having “large potential”. We give an informal overview here, and the formal definitions in Section 3.5 of the full version. We define a “startup event”  $\mathcal{E}_{\text{init}}(t_0)$  which occurs for some  $t_0$  with probability 1, and which guarantees very large potential at time  $t_0$  (governed by a constant  $C_{\text{init}}$ ). We divide the set of bins into *blocks*  $B_1, B_2, \dots$  of exponentially-growing length, and define a map  $\tau \mapsto \text{bins}(\tau)$  from times  $t_0 + \tau$  to blocks  $B_i$ . We define a constant  $\zeta > 0$ , and say that  $T$  is  *$t_0$ -jammed for  $\tau$*  if both  $Y^A$  and  $Y^B$  have at least  $\zeta|\text{bins}(\tau - 1)|$  potential at time  $t_0 + \tau - 1$ . (Observe that if every ball were stuck, then in expectation the balls in  $\text{bins}(\tau - 1)$  would contribute  $\lambda|\text{bins}(\tau - 1)|$  total potential, so jammedness says that on average the bins in  $\text{bins}(\tau - 1)$  are “almost full”.) We can now state Lemma 14.

**Lemma 14.** *Fix  $\lambda, \eta$  and  $\nu$  in  $(0, 1)$  and a  $(\lambda, \eta, \nu)$ -suitable send sequence  $\mathbf{p}$  with  $p_0 = 1$ . Let  $Y^A$  and  $Y^B$  be independent externally-jammed processes with arrival rate  $\lambda/2$  and send sequence  $\mathbf{p}$  and consider the two-stream externally-jammed process  $T = T(Y^A, Y^B)$ . Let  $t_0$  and  $\tau$  be sufficiently large integers. Then there is a coupling of*

- $T$  conditioned on  $\mathcal{E}_{\text{init}}(t_0)$ ,
- a sample  $\{Z_j^A \mid j \in \text{bins}(\tau)\}$  where each  $Z_j^A$  is chosen independently from a Poisson distribution with mean  $\lambda/(4p_j)$ , and
- a sample  $\{Z_j^B \mid j \in \text{bins}(\tau)\}$  where each  $Z_j^B$  is chosen from a Poisson distribution with mean  $\lambda/(4p_j)$  and these are independent of each other but not of the  $\{Z_j^A\}$  values

*in such a way that at least one of the following happens:*

- $T$  is not  $t_0$ -jammed for  $\tau$ , or
- for all  $j \in \text{bins}(\tau)$ ,  $|\text{stuck}_j^T(A, t_0 + \tau)| \geq Z_j^A$  and  $|\text{stuck}_j^T(B, t_0 + \tau)| \geq Z_j^B$ .

To prove Lemma 14 (or its full version Lemma 45 of the full version), we first formalise the idea that  $Y^A$  can act as a “relatively independent” source of noise to prevent balls in  $Y^B$  becoming unstuck (and vice versa). In Section 4.1 of the full version, we define a *random unsticking process*; this is an externally-jammed process in which balls become unstuck on sending based on independent Bernoulli variables, rather than the behaviour of other balls. In Lemma 51 of the full version,

we dominate the stuck balls of  $Y^A$  below by a random unsticking process  $R$  for as long as  $Y^B$  is “locally jammed”, and likewise for the stuck balls of  $Y^B$ .

In order to analyse the random unsticking process, we then make use of a time reversal, defining a *reverse random unsticking process*  $\tilde{R}$  in Section 4.2 of the full version which we couple to  $R$  using a probability-preserving bijection between ball trajectories set out in Lemma 47 of the full version. We then exploit the fact that balls move independently and the fact that the number of balls following any given trajectory is a Poisson random variable to prove Lemma 14, expressing the set of balls in  $\mathbf{stuck}_j^{\tilde{R}}$  as a sum of independent Poisson variables (one per possible trajectory) with total mean at least  $\lambda/(4p_j)$ . Unfortunately, the coupling between  $R$  and  $\tilde{R}$  does not run over all time steps — for example, since we need to condition on  $\mathcal{E}_{\text{init}}(t_0)$ , it certainly cannot cover times before  $t_0$  in  $R$ . As a result, we cannot analyse all trajectories this way; instead, we define a subset  $\text{Fill}_j^R(t)$  of balls in bin  $j$  of  $R$  at time  $t$  which are born suitably late in  $R$ , and analyse the corresponding set  $\text{Fill}_j^{\tilde{R}}(t)$  of balls in  $\tilde{R}$  (see Section 4.6 of the full version). We then prove in Lemma 48 of the full version that the variables  $|\text{Fill}_j^{\tilde{R}}(t)|$  for  $j \in \text{bins}(\tau - 1)$  are dominated below by a tuple of independent Poisson variables; this bound then propagates back through the couplings to a lower bound on  $|\text{Fill}_j^R(t)|$  in  $R$  and finally to the required bounds on  $|\mathbf{stuck}^{Y^A}(t)|$  and  $|\mathbf{stuck}^{Y^B}(t)|$  in  $T$  given by Lemma 14.

## 5 Future work

It is natural to wonder where the remaining difficulties are in a proof of Aldous’s conjecture. LCED sequences are quite a restricted case, and at first glance one might suspect they could be proved unstable by incremental improvements to Theorem 13, Corollary 9, and Lemma 7. Unfortunately, this is not the case — LCED send sequences can exhibit qualitatively different behaviour than the send sequences covered by these results.

Recall the definition of a  $(\rho, \mathbf{a}, g)$ -interleaved send sequence from Example 11. Let  $\mathbf{p}$  be such a send sequence, and let  $X$  be a backoff process with send sequence  $\mathbf{p}$  and arrival rate  $\lambda \in (0, 1)$ . Unless  $g$  is exponentially small, Lemma 7 does not apply to  $\mathbf{p}$  for any  $\lambda < 2\rho$ . Also, Theorem 8 (and hence Corollary 9) does not apply to  $\mathbf{p}$  for any  $\lambda \in (0, 1)$  whenever  $\rho$  is sufficiently small and  $\mathbf{a}$  grows suitably quickly. To see this, suppose that  $\rho$  is small and that  $\mathbf{a}$  grows quickly. Recall the definition of  $\mu_\tau(\mathbf{p})$  from Theorem 8. It is not hard to show (e.g. via union bounds) that  $\mu_\tau(\mathbf{p}) \leq \log_{1/\rho}(\tau) + O(1)$  for all  $\tau$  satisfying  $a_{2k} \leq \log_{1/\rho}(\tau) \leq a_{2k+1}/3$  for any  $k$ ; from this, it follows that

$$\sum_{\tau=0}^{\infty} \mu_\tau(\mathbf{p}) e^{-\lambda \mu_\tau(\mathbf{p})} \geq \sum_{\tau=0}^{\infty} \mu_\tau(\mathbf{p}) e^{-\mu_\tau(\mathbf{p})} = \infty \text{ for all } \lambda \in (0, 1).$$

Since Theorem 8 is both a necessary and a sufficient condition for a backoff process to send only finitely many times (see [19]), we can conclude that the methods of Kelly and MacPhee do not apply here.

As discussed in Section 4,  $\mathbf{p}$  is not LCED whenever  $g(j) = o(1)$  as  $j \rightarrow \infty$ . It is also easy to see that  $\mathbf{p}$  is LCED whenever  $g(j) = \Theta(1)$  and  $a_{2k+1} - a_{2k} = o(a_{2k+2} - a_{2k+1})$  as  $k \rightarrow \infty$ , so in this case our own Theorem 5 does not apply. To show that a simple generalisation of Theorem 13 will not suffice, we introduce the following definition.

**Definition 15.** *A send sequence  $\mathbf{p}$  has quiet periods if it has the following property for some  $\lambda \in (0, 1)$ . Let  $X$  be a backoff process with arrival rate  $\lambda$  and send sequence  $\mathbf{p}$ . Then with probability 1, there exist infinitely many time steps on which  $X$  has potential less than 1.*

Recall that the proof of Theorem 13 (as with Aldous’s result [4]) relies on proving that the potential of a backoff process increases to infinity over time, so it is unsuitable for send sequences

with quiet periods. Moreover, Corollary 9 is based on Kelly and MacPhee’s necessary and sufficient condition for a backoff protocol to have only finitely many successful sends (with probability 1). Backoff protocols with quiet periods have infinitely many successful sends, so cannot be proved unstable by the methods of Corollary 9. Finally, Lemma 7 handles many protocols with quiet periods (e.g.  $p_j = 2^{-2^j}$ ), but these protocols are *always* quiet, and have the property that they can be proved unstable by simple domination arguments (assuming that a ball always succeeds when it sends unless another ball is born at the same time). For many send sequences with quiet periods, these assumptions are not true.

The key problem is that if  $g(j) = 1/2$  (say),  $\rho$  is sufficiently small, and  $\mathbf{a}$  grows sufficiently quickly, then  $\mathbf{p}$  is very likely to have quiet periods. Indeed, in the externally-jammed process, for large values of  $k$  all bins  $1, \dots, a_{2k} - 1$  are likely to simultaneously empty (which happens with probability at least roughly  $2^{-a_{2k}}$  in the stationary distribution) long before bin  $a_{2k}$  fills with stuck balls and begins contributing significant potential (which will take time at least  $(1/\rho)^{a_{2k}}$ ). Heuristically, we would expect  $\mathbf{p}$  to alternate between “quiet periods” with very low potential and “noisy periods” in which potential first grows rapidly, then stays high for a very long time. Our current methods are incapable of handling this juxtaposition — in particular, since messages can easily escape from the system in quiet periods, any argument based around our ideas would need to bound the frequency and duration of quiet periods.

However, despite this, we believe we have made significant progress towards a proof of Aldous’s conjecture even for LCED sequences. Recall from Section 4 that the key part of our proof of Theorem 13 is Lemma 14, which gives a way of dominating the number of stuck balls in different bins below by an *independent* tuple of Poisson variables as long as overall potential is large. Importantly, the proof of Lemma 14 doesn’t actually rely on the fact that  $\mathbf{p}$  is suitable. Suitability is important to determining the parameters of the block construction, which is vital in applying Lemma 14 to prove Theorem 13, but the Lemma 14 proof goes through regardless of what these parameters are. As such, we believe a slightly more general version of Lemma 14 will be important even for LCED sequences as a tool for analysing “noisy periods”. The main remaining challenge is in showing that such “noisy periods” are common enough to outweigh the effect of “quiet periods” of very low potential.

## References

- [1] 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 Standard 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pages 1–3534, 2016.
- [2] Norman Abramson. THE ALOHA SYSTEM. In Norman Abramson and Franklin Kuo, editors, *Computer-Communication Networks*, pages 281–285. Prentice-Hall, 1973.
- [3] Hesham Al-Ammal, Leslie Ann Goldberg, and Philip D. MacKenzie. An improved stability bound for binary exponential backoff. *Theory of Computing Systems*, 34(3):229–244, 2001.
- [4] David Aldous. Ultimate instability of exponential back-off protocol for acknowledgment-based transmission control of random access communication channels. *IEEE Transactions on Information Theory*, 33(2):219–223, 1987.
- [5] Mark Allman, Vern Paxson, and Ethan Blanton. TCP congestion control. *RFC*, 5681:1–18, 2009.

- [6] Michael A. Bender, Tsvi Kopelowitz, William Kuszmaul, and Seth Pettie. Contention resolution without collision detection. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 105–118. ACM, 2020.
- [7] Haimin Chen, Yonggang Jiang, and Chaodong Zheng. Tight trade-off in contention resolution without collision detection. In Avery Miller, Keren Censor-Hillel, and Janne H. Korhonen, editors, *PODC '21: ACM Symposium on Principles of Distributed Computing*, pages 139–149. ACM, 2021.
- [8] Bogdan S. Chlebus. Randomized communication in radio networks. *CoRR*, abs/1801.00074, 2018. URL: <http://arxiv.org/abs/1801.00074>.
- [9] AWS documentation. Error retries and exponential backoff in AWS. URL: <https://docs.aws.amazon.com/general/latest/gr/api-retries.html>.
- [10] Google Drive documentation. Handle errors and improve performance. URL: <https://developers.google.com/drive/api/guides/handle-errors?hl=en#exponential-backoff>.
- [11] Anthony Ephremides and Bruce Hajek. Information theory and communication networks: an unconsummated union. *IEEE Transactions on Information Theory*, 44(6):2416–2434, 1998.
- [12] Leslie Ann Goldberg, Mark Jerrum, Sampath Kannan, and Mike Paterson. A bound on the capacity of backoff and acknowledgment-based protocols. *SIAM Journal on Computing*, 33(2):313–331, 2004.
- [13] Leslie Ann Goldberg and John Lapinskas. Instability of backoff protocols with arbitrary arrival rates, 2022. URL: <https://arxiv.org/abs/2203.17144>, doi:10.48550/ARXIV.2203.17144.
- [14] 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.
- [15] Jonathan Goodman, Albert G. Greenberg, Neal Madras, and Peter March. Stability of binary exponential backoff. *Journal of the ACM*, 35(3):579–602, 1988.
- [16] Albert G. Greenberg, Philippe Flajolet, and Richard E. Ladner. Estimating the multiplicities of conflicts to speed their resolution in multiple access channels. *Journal of the ACM*, 34(2):289–325, 1987.
- [17] Johan Håstad, Frank T. Leighton, and Brian Rogoff. Analysis of backoff protocols for multiple access channels. *SIAM Journal on Computing*, 25(4):740–774, 1996.
- [18] Maurice Herlihy and J. Eliot B. Moss. Transactional memory: Architectural support for lock-free data structures. In *Proceedings of the 20th Annual International Symposium on Computer Architecture, San Diego, CA, USA, May 1993*, pages 289–300. ACM, 1993.
- [19] Frank P. Kelly and Iain M. MacPhee. The number of packets transmitted by collision detect random access schemes. *Annals of Probability*, 15:1557–1568, 1987.
- [20] Robert Metcalfe and David R. Boggs. Ethernet: Distributed packet switching for local computer networks. *Communications of the ACM*, 19(7):395–404, 1976.



- [21] Jeannine Mosely and Pierre Humblet. A class of efficient contention resolution algorithms for multiple access channels. *IEEE Transactions on Communications*, 33(2):145–151, 1985.
- [22] Prabhakar Raghavan and Eli Upfal. Stochastic contention resolution with short delays. *SIAM Journal on Computing*, 28(2):709–719, 1998.
- [23] Ravi Rajwar and James R. Goodman. Speculative lock elision: enabling highly concurrent multithreaded execution. In Yale N. Patt, Josh Fisher, Paolo Faraboschi, and Kevin Skadron, editors, *Proceedings of the 34th Annual International Symposium on Microarchitecture, Austin, Texas, USA, December 1-5, 2001*, pages 294–305. ACM, 2001.
- [24] Lawrence G. Roberts. ALOHA packet system with and without slots and capture. *ACM SIGCOMM Computer Communication Review*, 5(2):28–42, 1975.
- [25] Devavrat Shah and Jinwoo Shin. Randomized scheduling algorithm for queueing networks. *Annals of Applied Probability*, 22:128–171, 2012.
- [26] Devavrat Shah, Jinwoo Shin, and Prasad Tetali. Medium access using queues. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011*, pages 698–707. IEEE Computer Society, 2011.