

Some New (And Old) Results on Contention Resolution

Leslie Ann Goldberg   

University of Oxford, UK

Abstract

This is an extended abstract of my talk at ICALP 2022, based on joint work with John Lapinskas.

2012 ACM Subject Classification Mathematics of computing

Keywords and phrases contention resolution, multiple access channel, randomised algorithms

Digital Object Identifier 10.4230/LIPIcs.ICALP.2022.3

Category Invited Talk

1 Extended Abstract

A *multiple access channel* is a shared channel which is used by processors to access a network, a cloud server, or another shared resource. The processors do not communicate except by listening to the channel. The mechanism of the channel is straightforward: If two or more processors send messages to the channel at the same time, then these messages collide, and are not delivered successfully. However, if during some time step exactly one processor sends a message, then this message is successfully delivered and the processor receives an acknowledgement of successful delivery. A *contention-resolution protocol* is a randomised algorithm that each processor uses to decide when to send its message to the channel (and when to wait because the channel is too busy).

We consider discrete-time contention resolution protocols in which processors communicate by sending discrete messages (packets) to a multiple access channel. During each step, new messages destined for the channel arrive at processors according to a probability distribution with an overall arrival rate λ . The whole process (consisting of the arrivals, the waiting messages, and the sends) can be viewed as a Markov chain. A contention-resolution protocol is *stable* [7] if this Markov chain is positive recurrent, which means that the process has a stationary distribution (so the expected build-up of waiting messages over time is bounded).

Two models of multiple access channels are the *queueing model* and the *queue-free model*. In the queueing model, there are N fixed processors. Each processor maintains a queue of messages which it is waiting to send – messages arrive at the tail of the queue and are sent from the head. The queues assist with stability, and are appropriate for applications with fixed networks. Dynamic networks (such as multiple access channels supporting cloud computing) are better captured by the queue-free model. In the queue-free model, we assume that processors join the network according to a probability distribution and that each processor has a single message to send. Thus, we identify the processor with its message – messages arrive at each step according to a Poisson distribution with rate λ and they stay in the system until they are successfully sent.

Two kinds of contention-resolution protocol are *full-sensing protocols* [12] which constantly listen the channel, obtaining partial information (such as which steps have no sends, which steps have exactly one send, which steps have a collision, or some combination of these) and *acknowledgement-based protocols*, which are appropriate for settings where this listening is not feasible. In acknowledgement-based protocols, the only information that a processor gets is whether its own sends are successful. A popular kind of acknowledgement-based protocol is



© Leslie Ann Goldberg;

licensed under Creative Commons License CC-BY 4.0

49th International Colloquium on Automata, Languages, and Programming (ICALP 2022).

Editors: Mikołaj Bojańczyk, Emanuela Merelli, and David P. Woodruff;

Article No. 3; pp. 3:1–3:3



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



a *backoff protocol*. A backoff protocol is associated with a send sequence $\mathbf{p} = p_0, p_1, \dots$ where each p_i is interpreted as a probability in $(0, 1]$. During a given time step, a processor that is trying to send a message that has already had i collisions flips a coin – with probability p_i , it sends. With probability $1 - p_i$, it instead waits for a later time step. One of the best-known backoff protocols is *binary exponential backoff*, which has $p_i = 2^{-i}$ and is the basis for Ethernet [11].

The talk will mention some important recent results [13, 14, 3, 4] about full-sensing protocols, however the main focus will be on acknowledgement-based protocols and specifically on backoff protocols. In the queueing model, binary exponential backoff is known to be stable for sufficiently small arrival rate λ [7, 1]. Unfortunately, this value of λ depends on N and binary exponential backoff is unstable if λ is sufficiently large [8]. In a breakthrough result, Håstad, Leighton and Rogoff [8] showed that there is a stable backoff protocol for every $\lambda \in (0, 1)$ – in particular *polynomial backoff*, with $p_i = i^{-\alpha}$ for some $\alpha > 1$, is stable.

In the queue-free model, it is conjectured [2] that no backoff protocol is stable for any positive arrival rate λ . This question was raised by MacPhee [10] and the same is conjectured for all acknowledgement-based protocols. The conjecture for backoff protocols is known to be true when $\lambda \geq 0.42$ and the conjecture for acknowledgement-based protocols is known to be true when $\lambda \geq 0.531$ [5].

The evidence for the full conjecture concerning backoff protocols is that a backoff protocol is known to be unstable for all positive arrival rates if its send sequence $\mathbf{p} = p_0, p_1, \dots$ decays smoothly. In particular, for the case where $1/p_j = o(c^j)$ for all $c > 1$, instability follows from work of Kelly and MacPhee [9] (which shows that in this case, with probability 1, only finitely many messages are successfully sent). In the case where $1/p_j = \Theta(c^j)$ for some $c > 1$ instability can be proved by generalising the proof of Aldous’s seminal instability result for binary exponential backoff [2]. In the case where the send sequences has an infinite subsequence p_{j_1}, p_{j_2}, \dots satisfying $1/p_{j_k} = \omega(c^{j_k})$ for all $c > 1$ is also easily dealt with. This appears as a lemma in [6] but the method was also known to the authors of [5].

The main issue which makes it difficult to prove the full conjecture is proving instability for backoff protocols that have a send sequence that decays at an inconsistent rate or doesn’t decay at all, with p_j s that jump back and forth between large and small values as j increase. The main part of this talk describes new work [6] with John Lapinskas that solves this problem except in a special case. This special case has the property that the send sequences alternates between p_j ’s that are at least $\Omega(1)$ and p_j ’s that are exponentially small (but no smaller). Moreover, the large p_j ’s have density $1 - o(1)$ in $\{p_1, \dots, p_n\}$ as $n \rightarrow \infty$. We therefore refer to them as “LCED” send sequences (for “largely constant with exponential decay”). These sequences are defined as follows.

► **Definition 1.** A send sequence \mathbf{p} is LCED (“largely constant with exponential delay”) 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 $j \rightarrow \infty$.

The main Theorem of [6] is as follows.

► **Theorem 2 ([6]).** Let \mathbf{p} be a send sequence which is not LCED. Then for every $\lambda \in (0, 1)$ the backoff protocol with arrival rate λ and send sequence \mathbf{p} is unstable.

The theorem has the following consequences.

► **Corollary 3** ([6]). For every $\lambda \in (0, 1)$ and every monotonically non-increasing send sequence $\mathbf{p} = p_0, p_1, \dots$, the backoff protocol with arrival rate λ and send sequence \mathbf{p} is unstable.

► **Corollary 4** ([6]). 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 protocol with arrival rate λ and send sequence \mathbf{p} is unstable.

The talk will conclude with a discussion of the prospects for proving the full conjecture.

References

- 1 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.
- 2 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.
- 3 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.
- 4 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.
- 5 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.
- 6 Leslie Ann Goldberg and John Lapinskas. Instability of backoff protocols with arbitrary arrival rates, 2022. doi:10.48550/ARXIV.2203.17144.
- 7 Jonathan Goodman, Albert G. Greenberg, Neal Madras, and Peter March. Stability of binary exponential backoff. *Journal of the ACM*, 35(3):579–602, 1988.
- 8 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.
- 9 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.
- 10 Iain M. MacPhee. *On Optimal Strategies in Stochastic Decision Processes*. PhD thesis, University of Cambridge, 1987.
- 11 Robert Metcalfe and David R. Boggs. Ethernet: Distributed packet switching for local computer networks. *Communications of the ACM*, 19(7):395–404, 1976.
- 12 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.
- 13 Devavrat Shah and Jinwoo Shin. Randomized scheduling algorithm for queueing networks. *Annals of Applied Probability*, 22:128–171, 2012.
- 14 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.