Brief Announcement: Deterministic Contention Resolution on a Shared Channel

Gianluca De Marco
Dipartimento di Informatica, University of Salerno, Italy
demarco@dia.unisa.it

Dariusz R. Kowalski
Department of Computer Science, University of Liverpool, UK
d.kowalski@liverpool.ac.uk

Grzegorz Stachowiak
Institute of Computer Science, University Wrocław, Poland
gst@cs.uni.wroc.pl

Abstract

A shared channel, also called multiple-access channel, is one of the fundamental communication models. Autonomous entities communicate over a shared medium, and one of the main challenges is how to efficiently resolve collisions occurring when more than one entity attempts to access the channel at the same time. In this work we explore the impact of asynchrony, knowledge (or linear estimate) of the number of contenders, and acknowledgments, on both latency and channel utilization for the Contention resolution problem with non-adaptive deterministic algorithms.

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases Shared channel, multiple-access channel, distributed algorithm

Digital Object Identifier 10.4230/LIPIcs.DISC.2018.44

1 Introduction

The formal model that is taken as the basis for theoretical studies is defined as follows, *cf.* the surveys by Gallager [9] and Chlebus [4] and the recent works [5, 7, 8]. A set of k *stations*, also called *nodes*, are connected to the same transmission medium (called a *shared channel*) and can communicate by transmitting and receiving messages on the shared channel in synchronous *rounds*. The stations have distinct ids in the range $[N] = \{0, 1, \ldots, N-1\}$. A *contention resolution* algorithm is a distributed algorithm that schedules the transmissions for each of the k stations possessing a packet, guaranteeing that every station eventually transmits individually (*i.e.*, without interfering with other stations) on the channel.

All the literature on this problem (with the exception of recent papers [3, 7, 8]) either assumed the (simplified) *static* situation in which the k stations are all activated at the very beginning (and therefore start simultaneously their transmitting schedules) [1, 11, 12, 13] or that the activation times are restricted to statistical or adversarial-queuing models [2, 6, 10]. In the dynamic scenario, considered in this paper, Bender et al. [3] gave a very efficient randomized adaptive algorithm with collision detection. In contrast, our work deals with deterministic non-adaptive algorithms, *i.e.*, protocols that are not using any channel feedback.

Inspired by the inherently decentralized nature of the multiple access model, and adopting the model from [8] developed in the context of randomized solutions, in this paper we focus on a more general *dynamic* scenario, in which the stations with packets could get awake (*i.e.*, start their local executions) in *arbitrary times*, *i.e.*, the sequence of activation times, also called a *wake-up schedule*, is totally determined by a worst-case adversary. This scenario,

32nd International Symposium on Distributed Computing (DISC 2018).
Editors: Ulrich Schmid and Josef Widder; Article No. 44; pp. 44:1–44:3
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

also called asynchronous, reflects the more realistic situation in which the stations are geographically far apart or totally independent (from themselves and/or from the scheduler which injects packets to the underlying communication protocol), and consequently each activation time is locally determined and cannot be known or predicted.

Although the communication is in synchronous rounds, we assume *no global clock* and *no system-based synchronization*: each station starts its local clock in the round when it wakes up, without knowing anything about the round numbers of other stations' clock. In the static model there is no distinction between the model with a global clock and that without it. Indeed, one can assume that a global clock is always available in the latter: all stations start simultaneously and therefore their clocks, will always tick the same rounds. In this sense, the dynamic model considered in this work is more general and challenging than the static one.

We measure the efficiency of a station in terms of its *latency*, *i.e.*, the number of rounds necessary for the station to transmit its packet successfully, measured since its activation time. The complexity of an algorithm, called an *algorithm latency* (or simply a *latency* if it is understood from the context) is defined as the maximum latency over all awaken stations. A *channel utilization*, sometimes called a *throughput*, is defined as the worst-case ratio between the contention size k (which corresponds to the absolute minimum number of rounds needed for all the awaken stations to transmit successfully to the channel) and the algorithm latency.

2 Our contribution

Our first result shows that if the number of contenders k (or a linear upper bound of it) is known and the stations switch-off after the acknowledgment of their successful transmissions, the channel admits efficient solutions: there exists a deterministic non-adaptive distributed algorithm working in $O(k \log k \log N)$ time. This is close to the known lower bound $\Omega(k \log(N/k))$. In terms of channel utilization, the algorithm achieves throughput $\Omega(1/(\log k \log N))$, which is close to the upper bound $O(1/\log(N/k))$.

In a nutshell, we first generate a randomized schedule that succeeds with very high probability and then we use the probabilistic method to show that a schedule allowing every station to transmit must exist. Since we know k, the schedule for any station can be organized in such a way that we start with a probability of transmission $O(1/k)$ and double it every $O(k \log N)$ rounds until the station transmits with constant probability. For any fixed station v, the rounds t at which v has a good (constant) probability of successfully transmission are those with the sum of all transmission probabilities at t being a constant. The main challenge is to show that there are sufficiently many rounds with such a favorable property, in order to get a sufficiently high probability of short latency allowing derandomization.

In the same settings but when k is unknown, we show an $\Omega(k^2/\log k)$ lower bound, which points out that the ignorance of contention k makes the channel nearly quadratically less efficient, even if the stations could switch-off after acknowledgments. In very broad terms, the proof is organized as follows. We start by defining a randomly generated wake-up pattern for the k stations. Then we prove that in such a (worst-case) random pattern no station is able to successfully transmit after $\Omega(k^2/\log k)$ rounds with a high probability. The probabilistic method is finally used to show that such a wake-up pattern exists.

In our final result we nearly match the above mentioned complexity (for unknown k) by presenting an upper bound of $O(k^2 \log N)$, which is achieved even if acknowledgments are not provided. In terms of channel utilization, the algorithm achieves throughput $\Omega(1/(k \log N))$, which is close to the upper bound $O((\log k)/k)$. The high level approach follows the lines of the first result. Here the additional challenge is the ignorance of parameter k, which complicates the design of the random schedule. In particular, in this case the transmission probabilities cannot depend on the unknown k. Thus, we let them start from a constant value and decrease (contrary to what we did when k was known). One of the main issues is the right choice of the decrease factor. On one hand it should be relatively fast in order to guarantee the sought latency. On the other hand, it cannot be too fast since avoiding collisions becomes harder in absense of switch-off's (due to no acknowledgements). We found out that starting from a constant probability and decreasing it every $O(\ln N)$ rounds from $1/\sqrt{j}$ to $1/\sqrt{j+1}$ for $j = 4, 5, 6, \ldots$, allows us to balance both challenges.

Surprisingly, our results imply that the knowledge of the contention size has an exponential impact on the deterministic utilization of an asynchronous channel, while it is known that for synchronized channels this feature does not influence asymptotically the channel utilization. The second implication concerns the impact of acknowledgments – our results exponentially improve deterministic channel utilization if (some estimate of) k is known, unlike the case of randomized algorithms where the corresponding improvement is only polynomial.

References

1. A. Fernández Anta, M. A. Mosteiro, and J. Ramon Mu noz. Unbounded contention resolution in multiple-access channels. *Algorithmica*, 67:295–314, 2013.
2. M. A. Bender, M. Farach-Colton, S. He, B. C. Kuszmaul, and C. E. Leiserson. Adversarial contention resolution for simple channels. In *Proceedings, 17th ACM Symp. on Parallel Algorithms (SPAA)*, pages 325–332, New York, NY, USA, 2005. ACM.
3. M. A. Bender, T. Kopelowitz, S. Pettie, and M. Young. Contention resolution with log-logstar channel accesses. In *Proceedings, 48th ACM Symp. on Theory of Computing (STOC)*, pages 499–508, Cambridge, MA, USA, 2016. ACM.
4. B. S. Chlebus. Randomized communication in radio networks. In P. M. Pardalos, S. Rajasekaran, J. H. Reif, and J. D. P. Rolim, editors, *Handbook on Randomized Computing*, pages 401–456. Springer, New York, NY, USA, 2001.
5. B. S. Chlebus, G. De Marco, and D. R. Kowalski. Scalable wake-up of multi-channel single-hop radio networks. *Theor. Comput. Sci.*, 615:23–44, 2016. `doi:10.1016/j.tcs.2015.11.046`.
6. B. S. Chlebus, D. R. Kowalski, and M. A. Rokicki. Adversarial queuing on the multiple access channel. *ACM Trans. on Algorithms*, 8:5:1–5:31, 2012.
7. G. De Marco and D. R. Kowalski. Contention resolution in a non-synchronized multiple access channel. *Theor. Comput. Sci.*, 689:1–13, 2017. `doi:10.1016/j.tcs.2017.05.014`.
8. G. De Marco and G. Stachowiak. Asynchronous shared channel. In *Proceedings, ACM Symp. on Principles of Distributed Computing, PODC 2017, Washington, DC, USA, July 25-27, 2017*, pages 391–400, 2017. `doi:10.1145/3087801.3087831`.
9. Robert G. Gallager. A perspective on multiaccess channels. *IEEE Trans. Information Theory*, 31(2):124–142, 1985.
10. L. A. Goldberg, P. D. MacKenzie, M. Paterson, and A. Srinivasan. Contention resolution with constant expected delay. *J. ACM*, 47(6):1048–1096, 2000.
11. A. G. Greenberg and A S. Winograd. Lower bound on the time needed in the worst case to resolve conflicts deterministically in multiple access channels. *J. ACM*, 32:589–596, 1985.
12. J. Komlós and A. G. Greenberg. An asymptotically optimal nonadaptive algorithm for conflict resolution in multiple-access channels. *IEEE Trans. on Information Theory*, 31:302–306, 1985.
13. D. Kowalski. On selection problem in radio networks. In *Proceedings, 24th ACM Symp. on Principles of Distributed Computing (PODC)*, pages 158–166, Las Vegas, NV, USA, 2005. ACM.