Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Contention resolution in a non-synchronized multiple access channel[☆,☆☆]

Gianluca De Marco[a,*], Dariusz R. Kowalski[b]

[a] *Dipartimento di Informatica, Università di Salerno, 84084 Fisciano (SA), Italy*
[b] *Department of Computer Science, University of Liverpool, Liverpool L69 3BX, UK*

ARTICLE INFO

Article history:
Received 20 July 2016
Received in revised form 15 May 2017
Accepted 19 May 2017
Available online xxxx
Communicated by K. Censor-Hillel

Keywords:
Multiple access channel
Contention resolution
Deterministic algorithms
Distributed algorithms

ABSTRACT

Multiple access channel is a well-known communication model that deploys properties of many network systems, such as Aloha multi-access systems, local area Ethernet networks, satellite communication systems, packet radio networks. The fundamental aspect of this model is to provide efficient communication and computation in the presence of restricted access to the communication resource: at most one station can successfully transmit at a time, and a wasted round occurs when more than one station attempts to transmit at the same time. In this work we consider the problem of contention resolution in a multiple access channel in a realistic scenario when up to k stations out of n join the channel at *different times*. The goal is to let at least one station to transmit alone, which results in successful delivery of the message through the channel.

We present three algorithms: two of them working under some constrained scenarios, and achieving optimal time complexity $\Theta(k\log(n/k)+1)$, while the third general algorithm accomplishes the goal in time $O(k\log n\log\log n)$.

1. Introduction

Multiple access channels are well-known communication media that form the basis to many extensively studied network systems such as Aloha multi-access systems, local area Ethernet networks, satellite communication systems, packet radio networks [1–3].

Preliminaries. The model that is at the basis of theoretical studies of the multiple-access channel can be defined in the following terms. We are given a set of stations each of them having a unique integer *ID* from the set $\{1,2,\dots,n\}=[n]$, for some integer n.

These stations communicate by sharing one communication channel. At each time slot any station can either transmit a packet of data to the channel or listen to the channel. Notice that parameter n also establishes an upper bound on the number of stations that can be attached to the channel.

[☆] A preliminary version of this paper has been presented at the 27th Annual IEEE International Symposium on Parallel and Distributed Processing, (IPDPS 2013), Cambridge, MA, USA, May 2013.

[☆☆] This work was supported by the Polish National Science Centre grant DEC-2012/07/B/ST6/01534.

[*] Corresponding author.

E-mail addresses: demarco@dia.unisa.it (G. De Marco), d.kowalski@liverpool.ac.uk (D.R. Kowalski).

http://dx.doi.org/10.1016/j.tcs.2017.05.014

Please cite this article in press as: G. De Marco, D.R. Kowalski, Contention resolution in a non-synchronized multiple access channel, Theoret. Comput. Sci. (2017), http://dx.doi.org/10.1016/j.tcs.2017.05.014

There is no central unit controlling the stations. A transmission is successful at a given time slot if and only if at that time slot there is only one transmitting station; in such case all stations get the message (including the one which transmitted the data, as it possesses it by default). If two or more stations transmit simultaneously in a given time slot, the messages collide and the transmission is not successful — *i.e.*, no station receives any of the transmitted messages.

One of the fundamental problems in this context is a *contention resolution* problem, also known as a *wakeup problem*, where at least one station among those who joined the computation on the channel has to transmit successfully (*i.e.*, alone) on the channel. Of course, the possibility of having collisions among the transmissions makes this task particularly difficult. An *algorithm* for the wake-up problem is a collection of n transmission schedules, one for each station, which eventually allows exactly one of the active stations to transmit on the channel, therefore waking up every other station. Once one of the active stations manages to send its message successfully on the channel, the message is heard by all other stations. The efficiency of the algorithm is measured by the time complexity, *i.e.* the number of time slots necessary to find the first time slot at which the transmission schedules allow exactly one station to send a message, counted from the first slot with at least one active station.

In the literature, there are many important assumptions that can be made on the model described above, each of them may have an impact on the time complexity. The first important assumption is about the amount of feedback received from the channel in the case of collision. Substantially, two different scenarios are studied in the literature. In the *collision detection* model, any station is able to hear an *interference noise* in the case of collision, allowing it to deduce the information that two or more stations tried to transmit in a given time slot. A weaker scenario, used in the present paper, assumes that no feedback signal is supplied by the channel in the case of collision, making it consequently impossible to distinguish between an occurred collision and the case where no station transmits. Another crucial assumption concerns whether all the active stations wake-up simultaneously or as in the more general case considered in this paper, the stations wake up, spontaneously and independently, in different time slots. Finally, the third central issue is the measurement of the elapsed time. Essentially the possibilities range between two extreme situations: the *globally* synchronous and the *locally* synchronous model. In the first model all the stations have access to a global clock. When a processor wakes up, it can see the current round number ticked by the clock. The other model is weaker. Each station has its own local clock, therefore, although the communication is synchronous, (*i.e.* all the clocks tick with the same rate) there is no global round number visible by every station. Each station can start counting the time from the time slot it wakes up, without knowing anything about the other round numbers. In this paper, the globally synchronous model is considered. The contention resolution problem considered in this paper can be formally defined as follows.

The contention resolution problem. We are given a multiple access channel where each station knows only its own *ID* from the known range $[n]$. Some number k of stations, with $1 \leq k \leq n$, can *spontaneously and independently* wake up, *i.e.*, each of them can start its activity at any moment. Let $s \geq 0$ be the first time slot such that some station is woken up. The problem is to assign transmission schedules to the stations, one per each station, such that there exists a time slot $t \geq s$ at which exactly one station (among the conflicting awaken stations at time t) transmits. We consider the worst-case scenario over all possible patterns of spontaneous wake up times of stations and measure the efficiency by the number of time slots between the first spontaneous wakeup and the first successful transmission, *i.e.*, $t - s$.

It must be stressed that most of the collision resolution research on multi access communication has its main motivation from the fact that very often most transmitters are inactive most of the time, while only a few are busy. If all n stations connected to the channel were active, one could apply one of the simplest schedules to resolve conflicts: the *time division multiplexing protocol*. This means that when there are n stations, n time slots will be needed. Of course, this becomes very inefficient when the maximum number k of possible awaken stations is very small compared to n. Moreover, given the fully distributed nature of the system, it is often unrealistic to assume that the stations can rely on the knowledge of the bound k or the starting time s, as both these parameters depend on actions taken independently by the participating stations without any sort of coordination. This paper focuses on the impact that the knowledge of the parameters k and s can have on the time complexity of the wake-up problem in the realistic scenario when the stations wake up spontaneously and independently in different time slots. Namely, we consider the following three scenarios.

Scenario A (s is known). Each station knows its own *ID* and the parameter n. In addition, every station knows the starting time s, *i.e.* the first time at which some station has woken up.

Scenario B (k is known). Each station knows its own *ID* and the parameter n. In addition, every station knows the maximum number k of possible awaken stations, but doesn't have any a priori knowledge about the wakeup times of other stations (included the starting time s).

Scenario C (neither s nor k is known). Every station knows only its own *ID* and parameter n.

Previous and related work. The collision resolution research for the multi access communication began in 1970 with Abramson's ALOHA network [1].

Komlós and Greenberg [22] were the first to consider the typical situation when a subset of k among n stations are awakened and have messages, and *all of them* need to be sent (successfully) to the multiple access channel as soon as