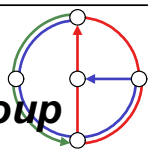
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing Group



AS 2007 Prof. Dr. R. Wattenhofer / Roland Flury / Stefan Schmid

Discrete Event Systems
Exercise 10

1 Competitive Analysis

In this exercise, we analyze algorithms for cellular networks such as GSM. In such networks, the area is segmented into cells, each of which containing a base station. Due to interference, base-stations in neighboring (adjacent) cells cannot use the same carrier frequencies, but frequencies may be reused in non-interfering cells (i.e., cells that are not neighboring). In this exercise, we use the idealized hexagonal grid for modelling these cells (cf Figure 1).

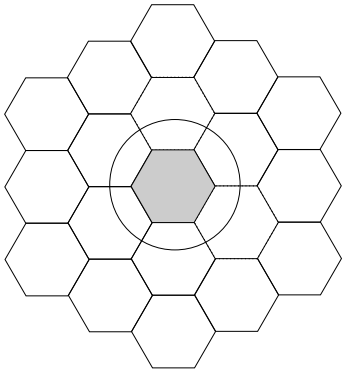


Figure 1: The hexagonal grid modelling the cells.

The number of frequencies in real GSM networks is limited. If there are more callers than channels, some calls must be rejected. In this exercise, we make the simplifying assumption that there is only a *single channel* shared by all base stations. That is, every base station can accept exactly one call. Furthermore, if a a call is accepted by a base station in a certain cell, no calls can be accepted in neighboring cells due to interference.

Assume that callers arrive in online fashion, that is, one after another in an input sequence σ. We need to accept or reject callers such that there is at most 1 caller in a cell and its 6 neighboring cells. The *benefit* of the algorithm is the number of calls we accept.

a) Assume that calls have *infinite duration*. Once a call is accepted, it remains forever. Describe a natural greedy algorithm for the problem. Is your algorithm ρ-competitive for any fixed constant ρ? If so, what is the value of ρ?

b) Assume that every call can have an arbitrary duration, but base stations are not allowed to preempt accepted calls. Propose a competitive online algorithm for this scenario.

c) Let us return to the case of calls with infinite duration. Do you think that your algorithm in a) is optimal? Is it possible to come up with a better online algorithm? Explain your decision.