Built-in Scheduling for Protocol Design

Endre Németh

The real-life protocols are the smallest logical units of communication. The protocols are designed as software elements from cooperating processes and some kind of scheduling is necessary among the processes. My goal is to describe and solve this scheduling problems. First I translate the protocol problems to the language of mathematics from the point of view of scheduling. The result of this translation is similar to the job-shop problems, but it differs sharply from it. The main differences are the following:

- it contains stochastic elements
- it has inventories.

I attempt to maximize the number of processed messages applying the best scheduling method.

The messages can be considered as jobs. The operation is the processing of the messages. A communication protocol is composed of the following processes:

- receivers,
- transmitters,
- controller and
- timer.

The paths of the messages among the cooperating processes are determined by constraints specified as state-transition rules. The receivers and transmitters are coders and decoders, the timer is a counter. The controller is a set of state-transition machines containing the state-transition rules. The scheduling is considered among the cooperating processes.

The scheduling problem considered among the cooperating processes is solved under the following main assumptions:

- no two operations of the same message may be processed simultaneously,
- each operation , once started, must be completed before another operation may be started on that process
- each message has m distinct operation, one on each processor
- the incoming messages are specified as arcs of two different protocol trees

We apply a heuristic algorithm to solve the problem. The validity of the algorithm is checked by simulation of a mobile protocol.