Model checking of multiuser power control algorithm for digital subscriber lines

Stéphanie Bay* — Thomas Brihaye† — Benoît Geller — Marc Jungers‡ — Nicolas Markey° — Ghassan Oreiby°**

* *Laboratoire SATIE (CNRS - ENS Cachan)*
61, avenue du Président Wilson – F-94235 Cachan Cedex
`bay@satie.ens-cachan.fr`

† *Institut de Mathématique – Université de Mons-Hainaut*
20, place du parc – B-7000 Mons
`Thomas.Brihaye@umh.ac.be`

** *U.E.I. ENSTA 32*
Boulevard Victor, 75739 Paris cedex 15
`geller@satie.ens-cachan.fr`

‡ *CRAN UMR 7039 CNRS - Nancy Université ENSEM*
2 avenue de la foret de Haye – F-54516 Vandoeuvre Cedex
`marc.jungers@cran.uhp-nancy.fr`

° *Lab. Spécification & Vérification (CNRS - ENS Cachan)*
61, avenue du Président Wilson – F-94235 Cachan Cedex
`{markey,oreiby}@lsv.ens-cachan.fr`

ABSTRACT. *The power control problem in data networks has become a critical optimization problem because nowadays the telecommunication resource is scarce and is shared everywhere by more and more users of greedy applications. This optimization problem is non-convex, because of the presence of these multiple users with their own objectives associated. The user satisfaction is represented by a quality of service, based on signal to interference ratio, and a cost term on user's power. A user is able to maintain the communication if its utility function reaches a lower threshold. In the static case, the water-filling provides a well-known and elegant solution, corresponding to a Nash equilibrium in the framework of Game Theory.*
To cope with time-varying number of users, or with the evolution of the channel characteristics, a new approach using an automata formulation of the game is presented

in this paper. This approach is not based on the research of an equilibrium, but rather on the verification that some quantitative properties hold in the game. For instance, we check if there exists a strategy for each user, which robustly guarantees the preservation of the telecommunication link when a perturbation appears. The introduction of a new user is a typical perturbation for the game. Futhermore this kind of approach is a powerful tool and helps at the design of algorithms controling each user's power allocation, especially the threshold levels. A simplified Uppaal-TiGA model, with two users using a single frequency channel, illustrates the robust behavior when a third player wants to begin a communication.

RÉSUMÉ. *Les systèmes de télécommunication connaissent un développement sans précédent. Ils sont devenus omniprésents, parfois indispensables, dans des domaines très variés, et la capacité des réseaux est souvent un facteur limitant. Dans ces conditions, le contrôle de la puissance d'émission de systèmes de communications est un problème critique. C'est un problème d'optimisation non-convexe, du fait de la présence de différents utilisateurs, chacun ayant des objectifs variés. Globalement, les objectifs se mesurent en termes de qualité de service calculée en fonction du rapport signal/bruit, et du coût d'utilisation calculé en fonction de la puissance d'émission nécessaire. Dans le cas stationnaire, le* water-filling *est une solution classique et élégante à ce problème, correspondant aux équilibres de Nash de la théorie des jeux.*

Dans cet article, nous proposons une nouvelle approche, basée sur une modélisation sous forme d'automates, pour traiter les cas non-stationnaires, où le nombre d'utilisateurs ou les caractéristiques du réseau peuvent varier dans le temps. Notre approche n'est pas fondée sur la recherche d'équilibre, mais plutôt sur la vérification de propriétés quantitatives sur le modèle. Nous vérifions par exemple si un utilisateur a une stratégie lui assurant de pouvoir émettre ses données malgré les perturbations possibles sur le réseau. Un exemple de perturbation est l'apparition d'un nouvel utilisateur sur le réseau. Cette approche nous permet d'étudier différents algorithmes de contrôle de puissance et d'obtenir de bonnes valeurs pour leurs paramètres. Nous illustrons notre approche à l'aide de l'outil Uppaal-TiGA, via *un petit exemple dans lequel un utilisateur peut communiquer de manière satisfaisante malgré les perturbations engendrées par deux autres utilisateurs.*

KEYWORDS: *timed games, model checking, verification, automata theory, digital subscriber lines, multiuser power control*

MOTS-CLÉS : *jeux temporisés, model-checking, vérification, automates, DSL, contrôle de puissance*

---

1. Introduction

Telecommunication networks have modified in depth our society in the recent years. It has become a technical challenge to let more and more people exchange more and more information anywhere and at any time. This means that in addition to the traditional perturbation of the Additive White Gaussian Noise (AWGN) (Proakis, 1995), there exists an other kind of random interference, due to the simultaneous existence of different users in the same frequency bands. This particularly holds in the framework of Digital Subscriber Lines (DSL) (Bingham, 2000), for which there are two kinds of crosstalk between adjacent lines: NEXT (Near-End Crosstalk) and FEXT (Far-End Crosstalk), as described in (ANSI Std, T1.E1.4/2003-210R1, 2003). Note that the same kind of problem arise in the case of mobile networks with for instance the Code Division Multiple Access (CDMA) interference.

In these systems, each user's performance not only depends on her own power allocation, but also on the power allocation of all the other users. Each user aims at maximizing her Signal to Interference-and-Noise Ratio (SINR), in order to maximize her flow rate. A first intuition to maximize her own SINR is to increase the corresponding power; however that implies a deterioration of the SINR of the other users. In addition the global power is limited, because the resources in energy are finite.

The required level of SINR depends on the application. There may not be any feasible power allocation to satisfy the requirements of all the users. The modification of the number of users may also change these constraints and corresponds to a hot topic in this framework. For example, an existing set of requirements can be satisfied, but when a new user is admitted into the system, there exists no more feasible power control solution by tightening of the constraint set.

An additional difficulty is the strongly nonlinear relation between the flow rate of the transmission and the SINR. The flow rate is a strictly increasing function of the SINR. In view of Shannon's second theorem (Shannon, 1949), the capacity of a channel (*i.e.*, the maximum flow rate) is given by a logarithm of the SINR. This kind of relation could be a plausible choice to link the current flow rate and SINR. To simplify, one maximizes the flow rate when the SINR is also maximized.

A first and intuitive approach to reach this goal is to design a centralized power control. That is design a power control allocation by a centralized agent, who gets all the information about the transmission system and about the SINR of each user. Then this centralized agent assigns each user's power level. However this approach is limited: in this case, the propagation delays could imply the instability of the control. Furthermore the formulation of the power control problem as a global optimization problem is too constraigning and could admit no feasible solution because of the complexity of the communication network. On the other hand, considering this problem as a complete decentralized optimization problem (that is, considering only single-user optimization problems) does not allow to depict the different crosstalk. The power control allocation problem necessitates to be modelized as a distributed system. Zander (Zander, 1992) and Yates (Yates, 1995) propose to consider power control allocation as a constrained optimization problem, where each user minimizes her transmission power level, while keeping her SINR larger than an upper threshold ensuring a good communication.

Game theory (Başar *et al.*, 1995) is a framework for analyzing the interaction of decision makers with conflicting objectives and limited resources. Recently in the literature, the power control problem has been viewed as a non cooperative game, in which each user attempts to minimize her own performance index in response to the actions of the other users. Different kinds of strategies are investigated, like Pareto strategy (Falomari *et al.*, 1998), or Stackelberg strategy (Başar *et al.*, 2002), or finally Nash strategy (Alpcan *et al.*, 2005). The literature providing a game-theoretic approach is rich in the telecommunication field (Ji *et al.*, 1998; MacKenzie *et al.*, 2001a; MacKenzie *et al.*, 2001b; Yu *et al.*, 2002; Lee, 2006; Subramanian *et al.*, 2005; Maheswaran *et al.*, 1998; Başar *et al.*, 2002; İmer *et al.*, 1999; Palomar *et al.*, 2004; Palomar *et al.*, 2003; Long *et al.*, 2007). A passivity approach should be noted (Fan *et al.*, 2006). The obtained results are noteworthy because conditions for the existence and the uniqueness of equilibria can be pointed out. This approach includes the optimization problem named *water-filling* (Chiang *et al.*, 2007; Palomar *et al.*, 2005), which is a formulation using the Karush-Kuhn-Tucker (KKT) conditions of optimality (Boyd *et al.*, 2004, page 245). An equivalence between water-filling and Nash equilibrium can be shown (Yu *et al.*, 2002).

In this paper a new approach is provided. A simple algorithm represented as an automata is applied to each user. The user increases her transmission power if her SINR is less than a lower threshold, or decreases it if her SINR is larger than an upper threshold, or maintain it if her SINR is included between them. The model-checking tool Uppaal-TiGA (Behrmann *et al.*, 2007) is used to study the resulting system, and to verify if one of the users can ensure correct communication quality whatever the other users do. Our objective is twofold: we first want to study our model for itself, and check if we can find correct values for the parameters in order to get a reliable system. But we also aim at evaluating the performances of Uppaal-TiGA on our models. For this second point, we will start with a small model and easy properties, and then study extensions of both. Our experiments were carried out on a 3GHz-CPU computer equipped with 2Go RAM, and Uppaal-TiGA v0.11 (`http://www.cs.aau.dk/~adavid/tiga/`).

The paper is organized as follows: in Section 2, the multiuser optimization problem is stated, the water-filling solution and the simplified decentralized algorithm are pointed out. Section 3 defines the models of timed automata and timed games we will use for modeling our systems. Section 4 depicts different timed games, with different levels of abstraction, modeling our telecommunication network. Finally, in Section 5, we present the results obtained with Uppaal-TiGA.

2. Problem Statement

2.1. *Problem formulation*

In this section, the problem of multiuser power allocation is formulated. In a multiuser telecommunication system, including N transmitters, each user or player ($k \in \{1, \ldots, N\}$) aims to transmit a signal x_k through a collection of channels. The outputs y_i ($i \in \{1, \ldots, N\}$) of the channels, or the received signals are obtained by filtering the send signals x_k with the impulse response h_{ik} of the channel. This filtering is represented with a convolution and is corrupted by adding a white gaussian noise n_i. In the frequency domain, where the Fourier transforms are noted with capital letter, the corresponding relations are

$$Y_i(f) = \sum_{k=1}^{N} H_{ik}(f) X_k(f) + N_i(f). \quad [1]$$

By considering that the signals x_k are independent, the interference theorem allows to obtain the power $\mathcal{P}(y_i, f)$ of the received signals.

$$\begin{aligned} \mathcal{P}(y_i, f) &= \sum_{k=1}^{N} |H_{ik}(f)|^2 \mathcal{P}(x_k, f) + \mathcal{P}(n_i, f), && [2] \\ &= \underbrace{\sum_{k=1, k\neq i}^{N} |H_{ik}(f)|^2 \mathcal{P}(x_k, f) + \mathcal{P}(n_i, f)}_{\text{Noise-Interference term}} + \underbrace{|H_{ii}(f)|^2 \mathcal{P}(x_i, f)}_{\text{Significative term}}. && [3] \end{aligned}$$

In this last expression [3], the Noise-Interference term and the significative term are stressed. The Noise-Interference term bring together the additional white gaussian noise power and the power of the others users' signals, corresponding to the different crosstalk (NEXT, FEXT). The significative term corresponds to the power of the signals, the user i wants to transmit. This leads to the definition of the Signal Interference Noise Rate (SINR).

Definition 1 *The SINR of the user i at frequency f is defined as the rate*

$$\gamma_i(f) = \frac{|H_{ii}(f)|^2 \mathcal{P}(x_i, f)}{\sum_{k=1, k\neq i}^{N} |H_{ik}(f)|^2 \mathcal{P}(x_k, f) + \mathcal{P}(n_i, f)}. \quad [4]$$

In view of Shannon's second theorem (Shannon, 1949), the capacity of a channel (*i.e.*, the maximum flow rate) is given by a logarithm of the SINR. This kind of relation could be a plausible choice to link the current flow rate and SINR. In the sequel of the paper, it is assumed that the flow rate $Q_i(f)$ at frequency f verifies the relation

$$Q_i(f) = \log\left(1 + \frac{\gamma_i(f)}{\Gamma}\right), \qquad [5]$$

where Γ is a positive parameter, called the *gap* of the channel. This parameter could be generally normalized to $\Gamma = 1$.

In practice, the transmission is digital, that the signal x_k are sampled to be transmitted. The frequency domain lies in $[0, F_s]$, where $F_s = \frac{F_{max}}{2}$ denotes the Shannon's frequency, related to the maximum frequency F_{max}, which could be transmitted in the channel. The global flow rate is given by a integral over all frequencies.

$$Q_i = \int_0^{F_s} Q_i(f)\mathrm{d}f = \int_0^{F_s} \log\left(1 + \frac{\gamma_i(f)}{\Gamma}\right)\mathrm{d}f. \qquad [6]$$

The purpose of each user is to maximize her own global flow rate Q_i. By considering that the power of the other users are fixed, an intuitive solution is to increase her own power $\mathcal{P}(x_i, f)$. However this simple solution necessitates an infinite resource in energy. That is obviously impossible. The optimization of the global flow rate should be designed by taking into account that the resources in energy are finite. The constraints could be formalized as

$$\frac{1}{F_s}\int_0^{F_s} \mathcal{P}(x_i, f)\mathrm{d}f \leq \mathcal{P}_{i,max}. \qquad [7]$$

Because there exist N objective values or criteria (the N global flow rates) to optimize, this is not possible to define a order relation with respect to a N-tuple. Consequently, the notion of *optimality* has no sense in the present case. This is necessary to call on *Game Theory* (Başar *et al.*, 1995; Engwerda, 2005) and the equilibrium associated with. The multiuser power control can be formulated as the following non-convex and constrained optimization problem, which is a constrained Nash strategy.

Definition 2 *Design the N-tuple of functions $\big(\mathcal{P}^*(x_1, f), \ldots, \mathcal{P}^*(x_N, f)\big)$, such that for all functions*

$$Q_i\big(\mathcal{P}^*(x_1, f), \ldots, \mathcal{P}^*(x_N, f)\big) \geq Q_i\big(\mathcal{P}^*(x_1, f), \ldots, \mathcal{P}(x_i, f), \ldots, \mathcal{P}^*(x_N, f)\big), \qquad [8]$$

under the constraints [7].

2.2. *Classical solving of the problem*

For the sake of clarity, the next notation is used

$$z(f) = \big(\mathcal{P}(x_1, f), \ldots, \mathcal{P}(x_N, f)\big). \qquad [9]$$

To solve the multiuser power control problem, a Lagrangian is defined for each user

$$\mathcal{L}_i(z(f), \dot{z}(f), f) = \lambda_0 \left(\log \left(1 + \frac{\gamma_i(f)}{\Gamma} \right) \right) + \lambda_1 \left(\mathcal{P}(x_i, f) - \mathcal{P}_{i,max} \right), \qquad [10]$$

where λ_0 and λ_1 are the *Lagrange multipliers* associated respectively with the global cost Q_i and the constraint [7]. The necessary conditions are given by the *Euler-Lagrange Equation*

$$\frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial \mathcal{L}_i}{\partial \dot{z}}(z(f), \dot{z}(f), f) \right) = \frac{\partial \mathcal{L}_i}{\partial z}(z(f), \dot{z}(f), f). \qquad [11]$$

In the particular case, where $\mathcal{L}_i$ is independent on $\dot{z}(f)$, the Euler-Lagrange Equation is degenerated and could be rewritten as

$$\frac{\partial \mathcal{L}_i}{\partial z}(z(f), \dot{z}(f), f) = 0. \qquad [12]$$

The equation [12] leads to the relation

$$\lambda_0 \frac{\partial}{\partial \mathcal{P}(x_i, f)} \left(\log \left(1 + \frac{\gamma_i(f)}{\Gamma} \right) \right) + \lambda_1 = 0. \qquad [13]$$

The Lagrange multipliers λ_0 and λ_1 cannot be simultaneously null. This implies that they are both different from zero and the optimization is *normal*. The relation [13] can be rewritten as

$$\Gamma \sum_{k=1, k\neq i}^{N} \frac{|H_{ik}(f)|^2}{|H_{ii}(f)|^2} \mathcal{P}(x_k, f) + \frac{\Gamma}{|H_{ii}(f)|^2} \mathcal{P}(n_i, f) + \mathcal{P}(x_i, f) = K_i, \qquad [14]$$

where K_i is a constant. The fact that this frequency-weighting sum of all power $\mathcal{P}(x_k, f)$ and $\mathcal{P}(n_i, f)$ is a constant independent on the frequency is known in telecommunication as *water-filling* (Chiang *et al.*, 2007; Palomar *et al.*, 2005). The determination of the constants K_i is an important field of research (Yu *et al.*, 2002; Shum *et al.*, 2007).

2.3. *Decentralized Algorithms*

In practice, the water-filling algorithm is not suitable, because of the difficulty of determining the constants associated with. In this paper, decentralized algorithms are preferred. It is assumed that each user has access to her own SINR. This availability allows each user to decide if she increases or decreases her own power allocation.

When the SINR is less than a low threshold, the flow rate is not enough to guarantee a correct communication, then the user should increase is power to maintain the transmission. When the SINR is greater than a upper threshold, the flow rate is sufficient to guarantee a correct communication, but the consumed energy is too important, then the user decreases her own power to preserve is energy.

In order to allow an automata-theoretic approach to cope with such a decentralized structure, it necessitates several simplifications:

– To avoid the dependency with respect to the frequency domain, only one frequency band is considered. This implies a simplification of the integrals defining the global flow rates and the constraints about the maximum power available.

– The set of values of each user's power is quantified. This leads to a quantification of the variations of the powers.

– The increasing or the decreasing of the powers are only allowed of one increment.

In addition such a decentralized structure permits to easily take into account a modification of the number of users. In the following section, the notion of automata is presented to be able to check the behavior of this model.

3. Timed automata and games

This section is devoted to the presentation of the models of timed automata, introduced originally in (Alur *et al.*, 1990), and their extension into timed games (Maler *et al.*, 1995; Asarin *et al.*, 1998).

3.1. *Timed automata*

Timed automata (Alur *et al.*, 1990; Alur *et al.*, 1994) are now considered as a natural, powerful and interesting model of real time systems. Roughly speaking a timed automaton is a finite automaton enriched with *clocks*. In this section we define and illustrate the notion of *timed automaton*. Let us first introduce some notations in order to define formally this notion.

We denote by $X = \{x_1, \ldots, x_n\}$ a set of n *clocks*. A *clock valuation* is a map $\nu\colon X \to \mathbb{R}^+$, where $\mathbb{R}^+$ is the set of nonnegative reals. Given a clock valuation ν, for $i = 1, ..., n$, we denote by ν_i the image of the clock x_i by the function ν, *i.e.* $\nu(x_i) = \nu_i$. Given a clock valuation ν, when no confusion is possible we also denote by ν the element of $(\mathbb{R}^+)^n$ given by $(\nu_1, ..., \nu_n)$. Given a clock valuation

ν and $\tau \in \mathbb{R}^+$, $\nu+\tau$ is the clock valuation defined by $(\nu_1+\tau, \ldots, \nu_n+\tau)$. A *guard* is any finite conjunction of expressions of the form $x_i \sim c$ or $x_i - x_j \sim c$ where x_i and x_j are clocks, $c \in \mathbb{N}$ is an integer constant, and $\sim$ is one of the symbols $\{<, \leq, =, >, \geq\}$. We denote by $\mathcal{G}$ the set of guards. Let g be a guard and ν be a clock valuation, notation $\nu \models g$ means that $(\nu_1, ..., \nu_n)$ satisfies g. A *reset* $Y \in 2^X$ indicates which clocks are reset to 0. We write AP for the set of *atomic propositions* that will be used for labeling the locations of our automata.

Definition 3 *A* timed automaton $\mathcal{A} = (L, X, \Sigma, E, \mathcal{I}, \mathcal{L})$ *has the following components:* (i) L *is a finite set of* locations*,* (ii) X *is a finite set of* clocks*,* (iii) Σ *is a finite set of* actions*,* (iv) $E \subseteq L \times \Sigma \times \mathcal{G} \times 2^X \times L$ *is a finite set of* edges*,* (v) $\mathcal{I} \colon L \to \mathcal{G}$ *assigns an* invariant *to each location, and* (vi) $\mathcal{L} \colon L \to 2^{AP}$ *is the* labeling *function.*

The semantics of a timed automaton $\mathcal{A}$ is given by a labelled transition system $T_{\mathcal{A}}$. A *state* of $\mathcal{A}$ is a pair $q = (l, \nu)$ such that $l \in L$ and $\nu \models \mathcal{I}(l)$. We let Q denote the set of all states.

We distinguish two kinds of transitions: *time-transitions* and *switch-transitions*:

– Given $q = (l, \nu)$ and $q' = (l', \nu')$ two states of $\mathcal{A}$, there is a *time-transition* in $\mathcal{A}$ between q and q' if there exists $\tau \in \mathbb{R}^+$ such that $l = l'$, $\nu' = \nu + \tau$ and $\nu + \tau' \models \mathcal{I}(l)$ for any τ', $0 \leq \tau' \leq \tau$. We denote this transition by $q \xrightarrow{\tau} q'$.

– Given $q = (l, \nu)$ and $q' = (l', \nu')$ two states of $\mathcal{A}$, there is a *switch-transition* in $\mathcal{A}$ between q and q' if there exists $e = (l, a, g, Y, l') \in E$ such that $\nu \models g$ and ν' is given by

$$\nu'_i = \begin{cases} 0 & \text{if } x_i \in Y \\ \nu_i & \text{if } x_i \notin Y. \end{cases}$$

We denote this switch-transition by $q \xrightarrow{e} q'$. To emphasize on the action a, we also use notation $q \xrightarrow{a} q'$ in this case, we use the notation $\mathsf{Action}(e) = a$.

We now define the (labelled) transition system $T_{\mathcal{A}}$.

Definition 4 *Given a timed automaton $\mathcal{A}$, the* (labelled) transition system associated with $\mathcal{A}$ *is given by* $T_{\mathcal{A}} = \big(Q, \Sigma \cup \mathbb{R}^+, \to\big)$ *where the transition relation is given by*

$$\to \; = \bigcup_{\tau \in \mathbb{R}^+} \xrightarrow{\tau} \cup \bigcup_{e \in E} \xrightarrow{e}.$$

Let $\mathcal{A} = (L, X, \Sigma, E, \mathcal{I}, \mathcal{L})$ be a timed automaton, q_1, q_2 and q_3 be three states of $\mathcal{A}$. If $q_1 \xrightarrow{\tau} q_2$, for some $\tau \in \mathbb{R}^+$, and $q_2 \xrightarrow{e} q_3$, for some $e \in E$, we shortly denote $q_1 \xrightarrow{\tau \cdot e}$ this sequence of two transitions. A finite or infinite run ρ is sequence of alternating transitions of the form:

$$\rho = q_1 \xrightarrow{\tau_1 \cdot e_1} q_2 \xrightarrow{\tau_2 \cdot e_2} \cdots \xrightarrow{\tau_k \cdot e_k} q_{k+1} \cdots .$$

We denote by $\mathrm{Run}_{\mathcal{A}}$ (resp. $\mathrm{Run}^{f}_{\mathcal{A}}$) the set of runs (resp. finite runs) of $\mathcal{A}$.

Let us now give an example of timed automaton modelling a simple real life system. We model a mouse producing simple or double *click* inspired from (Krichen *et al.*, 2005).

Example 1 *In order to model a mouse producing simple or double* click*, the timed aspect is essential. Indeed the mouse produces a* double-click *when the button is pressed twice quickly enough; otherwise it just produces two* simple-click*. In this example we assume that the mouse produces a* double-click *when the button is pressed twice within* 1 *time unit. This situation can be modelled by the one clock timed automata depicted on Fig. 1 together with the set of atomic proposition AP given by* {Idle,Simple,Double}*.*

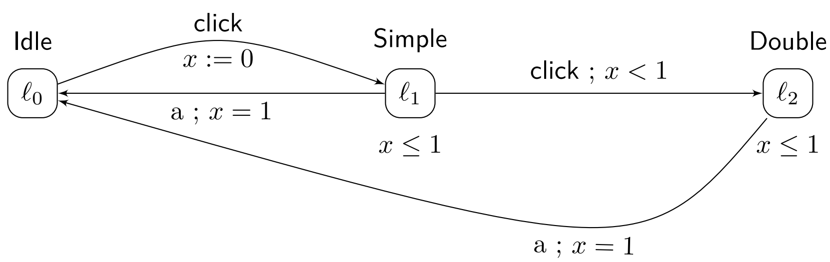

Figure 1: A timed automata modelling a simple-, double-*click* mouse

One of the most famous results on timed automata is the decidability of the reachability problem. More precisely, the reachability problem has been shown PSPACE-complete in (Alur *et al.*, 1994). The key ingredient to prove this result is the construction of a finite graph, namely the *region graph*, which is shown to be *time-abstract bisimilar* to the original timed automaton. In particular, the reachability problem on timed automaton reduces in the reachability problem on this finite region graph. Many other interesting model-checking-related problems have been shown decidable on timed automata (Alur *et al.*, 1993; Alur *et al.*, 1996). In parallel with these theoretical results, efficient verification tools such as Uppaal, Kronos, ... have been implemented and successfully applied to industrial relevant case studies (Daws *et al.*, 1996; Larsen *et al.*, 1997; Behrmann *et al.*, 2004).

3.2. *Timed games*

Timed automata are models for as closed systems, where every transition is controlled. If we want to distinguish between actions of a *controller* and actions of an *environment*, or more generally if we need to model multi-agent systems, we have to consider *games* on timed automata, also known as *timed games* (Maler *et al.*, 1995; Asarin *et al.*, 1998). In this context, we can ask if the controller can force the environment to update the control of the automaton in a way to achieve a given objective.

Definition 5 *Let $\mathcal{A} = (L, X, \Sigma, E, \mathcal{I})$ be a timed automaton. We say that $\mathcal{A}$ is a* timed game*, if the set of action Σ contains a particular action denoted u.*

In this context, the transitions labelled with u are called the *uncontrollable transitions*. They represent the set of actions available to the environment. The other ones are called the *controlled transitions*. We denote by Σ_c the set of actions $\Sigma \setminus \{u\}$.

Before giving the semantics of timed games, let us first explain it intuitively.

Let $\mathcal{A} = (L, X, \Sigma, E, \mathcal{I})$ be a timed game. The game is played by two players, *Player* 1 (the *controller*) and *Player* 2 (the *environment*). At any state q, Player 1 picks a time τ and an action $a \in \Sigma_c$ such that there is a transition $q \xrightarrow{\tau \cdot e} q'$ with $\mathsf{Action}(e) = a$. Player 2 has two choices:

– either she can decide to wait for time τ and execute a transition $q \xrightarrow{\tau \cdot e} q'$ proposed by Player 1,

– or she can wait for time τ', $0 \leq \tau' \leq \tau$, and execute a transition $q \xrightarrow{\tau' \cdot e'} q''$ with $\mathsf{Action}(e') = u$.

The game then evolves to a new state (according to the choice of Player 2) and the two players proceed to play as before.

Notice that, in the definition of a timed game, it is implicitly supposed that Player 1 can always formulate a choice (τ, a) in any reachable state q of the game.

We will now formalize the semantics through the concept of *strategy*.

Definition 6 *A* (Player 1) strategy *is a function*

$$\lambda \colon \mathit{Run}^f_{\mathcal{A}} \mapsto \mathbb{R}^+ \times \Sigma_c.$$

Before defining the notion of a winning strategy, we need to define several other notions. We say that a run ρ is *maximal* if it is either infinite or ending in a deadlock. An objective Ω of a timed game is a subset of the runs of $\mathcal{A}$. Let ρ be a run of the form $q_1 \xrightarrow{\tau_1 \cdot e_1} q_2 \xrightarrow{\tau_2 \cdot e_2} \cdots \xrightarrow{\tau_k \cdot e_k} q_{k+1} \cdots$, we denote by ρ_i the prefix of ρ ending in q_i. Given a strategy λ and a run ρ, we say that ρ is *played according to* λ if for every i, if $\lambda(\rho_i) = (\tau'_i, a_i)$, then either $\tau_i = \tau'_i$ and $\mathsf{Action}(e_i) = a_i$, or $\tau_i \leq \tau'_i$ and $\mathsf{Action}(e_i) = u$. We denote by $\mathsf{Outcome}(\rho, \lambda)$ the set of *maximal* runs extending ρ and played according to λ. Given a state q, a strategy λ and an objective Ω, we say that the strategy λ *is winning for the objective* Ω *from* q if $\mathsf{Outcome}(q, \lambda) \subseteq \Omega$.

In order to illustrate the more complex notion of timed game, let us give a simple example.

Example 2 *Let us first imagine a simple process controlled by a user which can work in two different modes:* Low *or* High*. One can imagine that the* High *mode is more efficient but energy consumings. We impose two constraints on this system. The first constraint is that the user is not allowed to switch too quickly from one mode to another, the second constraint impose that the user really use the two different modes. More precisely, when she enters the* Low *(resp.* High*) mode, she has to stay between* m_{L} *and* M_{L} *(resp.* m_{H} *and* M_{H}*) time units in this mode. This single user process can be modeled by the timed automaton of Fig. 2.*

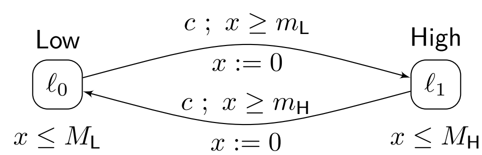

Figure 2: A system with High and Low modes

Now imagine that several copies of the our process run in parallel on the same device. Each process is controlled by a different user. In particular, each user can choose independently when to run her own process on the Low *(resp.* High*) mode. Let us also assume that the device can not afford that all the process run in the* High *mode for a too long time. More precisely, the device is blocked if all the process stay in the* High *mode for more than nine time units. In this framework, one can naturally ask if a given user has a way of controlling her own process in order to ensure that the device will not reach the blocking state, whatever the other users will do. This situation can be modelled via timed games.*

The case of two users, with some given constants for the guards and invariants, is modelled in Fig. 3. In this figure, the locations ℓ_0 *to* ℓ_3 *model the different modes of the two users and location* ℓ_4 *is the blocking location. In this example, we see the first user as the controller and the second user as the environment. One can show that the controller has no strategy to avoid the blocking location from* $(\ell_0, 0, 0, 0)$*. Even in this simple example, the non-existence of a winning strategy is not obvious.*

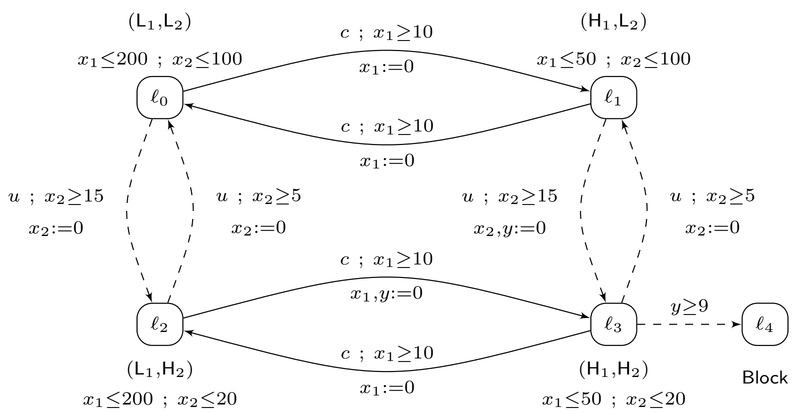

Figure 3: Example of a timed game

3.3. *Timed* ATL

TATL (Henzinger *et al.*, 2006; Brihaye *et al.*, 2007) is a temporal logic used to specify timed requirements when the model consists of a multiplayer system, it extends the untimed logic ATL (Alur *et al.*, 2002) (alternating-time temporal logic). Formulae written with TATL may be later used as an input along with the model in a model-checking tool to check whether our system satisfy the requirements or not.

Definition 7 *The syntax of* TATL *is defined by the following grammar:*

$$\begin{aligned} \mathsf{TATL} \ni \varphi_s, \psi_s \quad &::= \quad \top \mid P \mid \neg\varphi_s \mid \varphi_s \vee \psi_s \mid \langle\!\langle A \rangle\!\rangle\, \varphi_p \\ \varphi_p \quad &::= \quad \varphi_s\, \mathbf{U}_{\sim c}\, \psi_s \mid \varphi_s\, \mathbf{R}_{\sim c}\, \psi_s \end{aligned}$$

with $P \in AP$*,* $A \subseteq \{1,2\}$*,* $\sim \in \{<,\leq,=,\geq,>\}$*, and* $c \in \mathbb{N}$*.*

The *strategy quantifier* $\langle\!\langle A \rangle\!\rangle\, \varphi_p$ denotes the fact that coalition A has a strategy whose all outcomes satisfy φ_p. Formulas φ_p are called “path formulas”, because they impose requirements on the paths (or executions) of the game. We only give the semantics on an example: an outcome satisfies the property $a\,\mathbf{U}_{\leq 5}\, b$ if it reaches a location labelled with b within 5 time units, and only visits states labelled with a (modality $\mathbf{U}$ is read “until”).

In the sequel, we use only two simpler modalities, denoted by $\mathbf{F}_{\sim c}\, \psi_s$ and $\mathbf{G}_{\sim c}\, \psi_s$. The former, read “eventually”, is an abbreviation of $\top\,\mathbf{U}_{\sim c}\, \psi_s$; that is, $\mathbf{F}_{\leq 5}\, b$ means that the outcome visits a b-state within 5 time units. A simpler version with no timing constraint is also available: $\mathbf{F}\, b$ simply means that a b-state will eventually be visited. Modality $\mathbf{G}$ is the dual, and is read "always": $\mathbf{G}_{\leq 5}\, a$ means that, in the first 5 time units of the execution, only a-states are visited. The unconstrained version $\mathbf{G}\, a$ states that only a-states are visited all along the execution.

The tool Uppaal-TiGA (Behrmann *et al.*, 2007) used for the verification in this paper uses a fragement of TATL. The only modalities that can be used are $\langle\!\langle A \rangle\!\rangle\, \mathbf{F}$ and $\langle\!\langle A \rangle\!\rangle\, \mathbf{G}$, without nesting, but the clocks of the model can be used within the formula.

4. Modeling power allocation in communicating systems

We now present our models. We begin with a model made of two simple communicating devices. This will be our starting point, for which we will show that Uppaal-TiGA answers quite quickly. We will then extend this model in two directions, and see that the limits of Uppaal-TiGA are rapidly reached.

4.1. *Simple case: two systems, one mode of operation*

We model a network of communicating systems as a timed game: each system is a timed automaton, one of them being the protagonist. They all follow the same rules, namely:

– the power allocation is discretized, ranging from 0 to some maximal value (25 in our examples);

– when turned on, each system sets its power to 15;

– at any time, each system can adjust its power allocation in order to maintain its SINR between two values (15 to 20 in our examples), but it *has to* lower it as long as it exceeds a given threshold.

– each system can be turned on and off at any time, except that it must stay in one or the other state for a certain delay before changing anew.

Fig. 4 depicts[1] our model. In our examples below, this model will have identity $\mathtt{id} = 0$, and is the protagonist. The minimum `delay` is 1 between any two events in that system, while `ondelay` and `offdelay` equal 5 time units.

The SINR for the protagonist is $(\mathtt{A} \cdot \mathtt{p}[0])/(\mathtt{B} + \mathtt{C} \cdot \mathtt{p}[1])$, with $\mathtt{A} = 100$, $\mathtt{B} = 10$ and $\mathtt{C} = 2$. State `On` has three self-loops: the leftmost one increases the power by one unit, but is possible only if the SINR is below a certain threshold (represented by $\mathtt{m}[0]$ here). The rightmost selfloop allows the system to decrease its power in case it is too high. The third self-loop also decreases the power

1. Let us notice that our model contains guards of the form $(\mathtt{A} \cdot \mathtt{p}[0]) > (\mathtt{B} + \mathtt{C} \cdot \mathtt{p}[1])$. This is not a violation of the syntax of timed automata since the variables $\mathtt{p}[0]$ and $\mathtt{p}[1]$ are not clocks variables but discrete variables used to represent succinctly numerous locations.

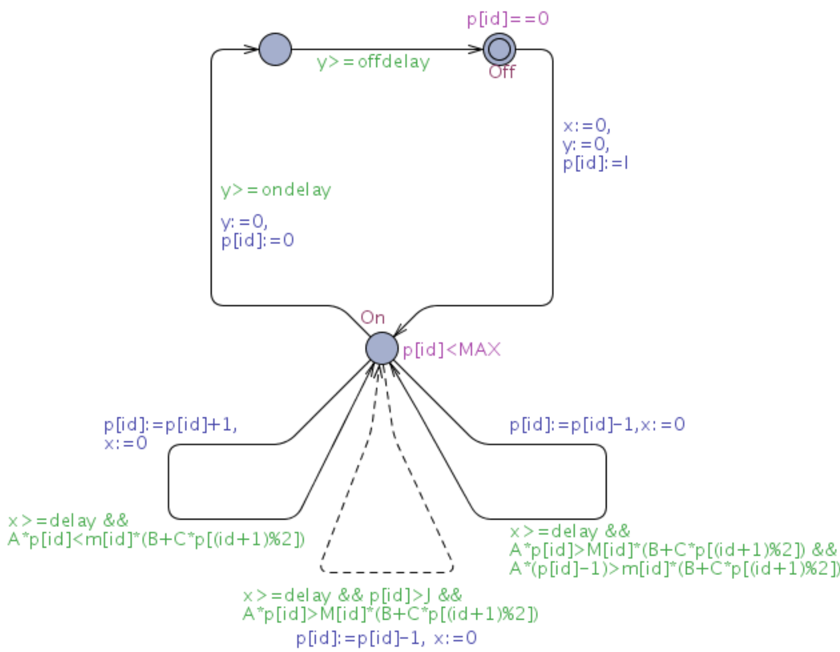

Figure 4: A model of a communicating system

allocation if it is too high, but this self-loop in *uncontrollable* (dashed transition on Fig. 4), modeling the fact that the system *has to* lower its power allocation if the SINR is high enough. The model for the opponent is symmetric.

Thanks to the simulation tool provided by Uppaal-TiGA, we can check that this system behaves as we expect. For instance, if the protagonist turns on, its power decreases as long as the antagonist is off, because of the uncontrollable self-loop. The system then reaches a stable state, where the protagonist sends her information with power equal to $J = 3$. The only allowed transition is then to turn off, or to decrease power some more.

If the antagonist comes up, her power is set to 15, and the SINR for the protagonist is modified, enabling the transition for increasing her power allocation. At the same time, the antagonist *has to* decrease her power allocation. Again, this converges to some "equilibrium" where both systems have reasonable values allowing them to send their data.

We then want to model the fact that establishing the communication requires a reasonably high value for the SINR, and more importantly, that communication is broken if the SINR goes below a given threshold. This is modeled through an extra module playing the role of an *observer* (in the sense that it should not interfere with the behaviour of the initial system), which will have two modes, representing the fact that the communication is established or broken, respectively. Our observer is depicted on Fig. 5. This simple observer

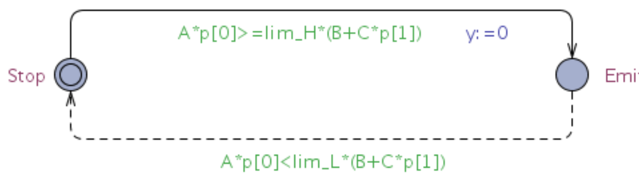

Figure 5: An observer detecting if the communication is established

can go the state `Emit` if the SINR is sufficiently high, but will go back to `Stop` as soon as the SINR will be too low. The first transition is controllable while the second one is uncontrollable, modeling the opposite goals of the protagonist and the antagonist.

We now have a reasonably complete model for our system, and are ready for checking several properties it should satisfy. Our main aim is to check that the protagonist has a strategy to enter location `Emit`. This is expressed through the following ATL formula:

$$\langle\langle A \rangle\rangle\, \mathbf{F}\, \texttt{Emit} \qquad \text{[Prop1]}$$

which reads "Player A (the protagonist) has a strategy to force the system reach location `Emit`". In the tool Uppaal-TiGA, this is expressed as a control requirement:

$$\texttt{control} : \texttt{A} <> (\texttt{Observer.Emit})$$

In our model, this property is satisfied if constants have reasonable values. The next step is to check that it is possible to stay in `Emit` for a sufficient delay. The classical way of expressing this requirement in timed ATL is through the following formula:

$$\langle\langle A \rangle\rangle\, \mathbf{F}\, (\, \langle\langle A \rangle\rangle\, \mathbf{G}_{\leq 15}\, \texttt{Emit}) \qquad \text{[Prop2]}$$

This reads: "player A has a strategy for reaching a state from which she can enforce the game to stay in location `Emit` for at least 15 time units".

Unfortunately, Uppaal-TiGA is not able to handle such formulas. Still, it allows to use the clocks of the model in the formulas. That's the reason why our `observer` module resets a clock y when entering location `Emit`: this clock will be used to measure the time elapsed in location `Emit`. The requirement we will check thus writes:

$$\texttt{control} : \texttt{A} <> (\texttt{Observer.Emit \&\& Observer.y} > 15)$$

This time, the requirement is too strong (except for certain values of the parameters, namely when $\texttt{lim}_\texttt{L}$ is small enough or if `J` is high enough[2]): in many situation, it can happen that the protagonist will be "surprised" by the antagonist coming up, resulting in a broken communication. This led us to refine our observer, allowing the SINR to be low for a finite amount of time. This is modeled in our new observer, depicted on Fig. 6. The parameter `timeout` represents the maximal delay before the communication is really broken. In that setting, being in location `Emit2` is not a problem as long as the protagonist has a way to go back to `Emit` before the timeout expires. The property to check is that the observer can be forced to stay in one of the locations `Emit` or `Emit2` for a sufficiently long time:

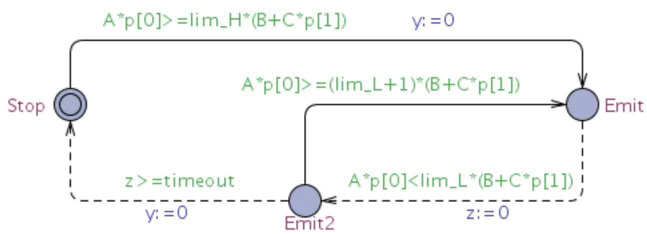

Figure 6: A more permissive observer

$$\langle\!\langle A \rangle\!\rangle\, \mathbf{F}\, [\, \langle\!\langle A \rangle\!\rangle\, \mathbf{G}_{\leq 15}\, (\texttt{Emit} \vee \texttt{Emit2})] \qquad \text{[Prop3]}$$

Again, we use the same trick as above in order to encode this property in Uppaal-TiGA :

$$\texttt{control} : \texttt{A} <> ((\texttt{Observer.Emit} \ \texttt{||} \texttt{Observer.Emit2}) \texttt{\&\&}\ \texttt{Observer.y} \geq 15)$$

4.2. *Extended case: two systems, two modes of operation*

An interesting extension of our model consists in having several modes of operation: indeed, depending on the data being sent, the system may need to have a better flow rate. We thus allow the system to switch to a mode where the range of the SINR is higher. The modification to the models is small: we simply add two loops on location `On`, each one corresponding to one of the two possible modes. The new model is depicted in Fig. 7. Since the observer

2. There is another reason why such a requirement could fail to hold: if the antagonist has a way for *blocking time* (a.k.a. *Zeno* strategy): this could happen if the antagonist had a way to execute infinitely many transitions in a finite delay. This is forbidden in our model thanks to the `delay` parameters being positive. We could check that, in the case where the `delay` parameters are set to 0, the property is always false.

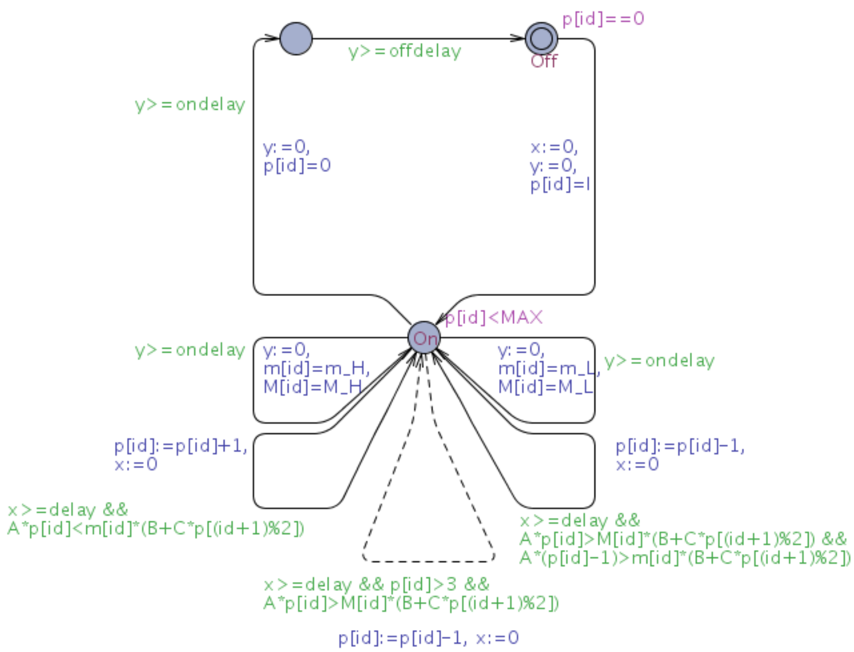

Figure 7: A model with two operating modes

depends only on the values of the `p[id]` variables, we use the same as the previous case.

Note that while the changes are small, the computation time was already much increased. See Section 5 for the list of our results.

4.3. *Extending to three players*

Another obvious extension of our model consists in adding extra players. Indeed, it might be the case that our protagonist can react quickly enough against the arrival of one single player, but the question is now whether two antagonists could "cooperate" in order to break the communication of the protagonist.

The SINR for the protagonist is now $(\mathtt{A} \cdot \mathtt{p}[0])/(\mathtt{B} + \mathtt{C} \cdot \mathtt{p}[1] + \mathtt{D} \cdot \mathtt{p}[2])$, with similar values as previously: $\mathtt{A} = 100$, $\mathtt{B} = 10$, $\mathtt{C} = 2$ and $\mathtt{D} = 2$. Hence, we replace the SINR in the previous model with the new one. The same applies to the observer as well. Then, the whole system consists now to 3 players: one protagonist and two opponents. We won't display the model as it is a direct adaptation of the previous ones.

Again, this addition dramatically increases the complexity of the problem, and Uppaal-TiGA could not achieve the verification of our formulas. We managed to get partial results, where the values of the parameters have been simplified. We report on this in the next section.

5. Verification with Uppaal-TiGA

We now explain the results we got when running the model-checker Uppaal-TiGA (Behrmann *et al.*, 2007) on our models for the properties we are interested in.

5.1. *Model with two players and one operating mode*

For this model, the answer to query [Prop1] is almost immediate, and the answer is positive: the protagonist has a strategy in order to reach location `Emit`. However, as already mentioned earlier, if we set `ondelay` and `offdelay` to zero, the formula does not hold anymore, as the antagonist has a “strategy” that prevents the protagonist to play any move: it consists in always switching on and off without any delay.

In the sequel, we will keep positive values for the delays.

We now check if Property [Prop2] holds. When setting both $\mathtt{lim}_\mathtt{L}$ and $\mathtt{lim}_\mathtt{H}$ to 15 (with $\mathtt{I} = 15$ and $\mathtt{J} = 3$) the property fails to hold, which is checked by Uppaal-TiGA within 30 seconds. Of course, if we lower the value of the bound $\mathtt{lim}_\mathtt{L}$ or if we increase the value of `J`, the property can be recovered. Also, we can prove that it is possible to stay in location `Emit` for at least 5 time units: this is because the power is set to a high value when beginning the communication, and decreases slowly. During that time, the protagonist can stay in location `Emit`.

We can also allow the SINR to go below the bound for a limited delay, as encoded using the enhanced observer and Property [Prop3]. Again, we tried our experiments with both $\mathtt{lim}_\mathtt{L}$ and $\mathtt{lim}_\mathtt{H}$ set to 15, and with $\mathtt{I} = 15$ and $\mathtt{J} = 3$. With a timeout value of 3 time unit, Uppaal-TiGA has be able to prove that Property [Prop3] holds. The computation time was around 3 minutes. Again, we could check that the values of the parameters are tight in this case, as the property fails to hold if we have a smaller timeout.

5.2. *Model with two players and two operating modes*

In our first extension, where each system has two modes, Property [Prop1] is again quickly shown to hold. With our values of the parameters, the extra mode does not provide a way for the protagonist to prevent the communication from being broken at some point: Property [Prop2] still fails to hold with our values. What is more surprising is this slight extension makes verification much harder: it now takes 36 minutes for Uppaal-TiGA to check the second property.

Checking that Property [Prop3] holds in our model was even harder: it took Uppaal-TiGA more than 6 hours and 30 minutes to check this property. In our example, lowering the upper bound for `p[id]` does not help since "high" values are unreachable.

The fact that property [Prop3] holds is not surprising by itself: the cases when the communication is broken occur, roughly, when the antagonist initiates a new communication while the protagonist has low power. The new mode allows the protagonist to have higher power, which makes Property [Prop3] hold "more robustly".

5.3. *Model with three players*

The size of the system dramatically increases with the addition of an extra player: while the graphical representation of Uppaal-TiGA is very succinct thanks to the use of variables, each location of our small models represent in fact a set of 25 locations (because we have discretized our model with that maximal value).

Still, the first property is checked quite easily in case there is only one mode of operation. By simplifying our model (lowering the upper bound of `p[id]` to 16, which does not change the satisfaction of our property, merging both `Off` locations, and setting all delays to 1), we were able to prove that Property [Prop2] fails to hold. This simplified model could also be used for proving that Property [Prop3] fails with a timeout value of 3. Due to the third player, the timeout has to be raised to 4 in order to satisfy Property [Prop3]. Checking this latter property took 17 hours.

Apart from those simplified cases, Uppaal-TiGA has not been able to check any other properties of our three-player model. The model obtained by combining both extensions (2 modes and 3 players) is too large to be handled by the tool, even with our simple property [Prop1].

6. Conclusion

In this paper, we have proposed a timed-automata-based study of the problem of power allocation in telecommunication networks. The interest of our approach is twofold: on the one hand, this approach using timed automata and timed games is original in the field of telecommunications, and seems promising in view of our first results: instead of studying the equilibria that can be reached in telecommunication networks, it focuses on the perturbations brought by the arrival of new senders on the network.

On the other hand, our approach shows the performances and limitations of the state-of-the-art tool Uppaal-TiGA : the verification of timed games has very high theoretical complexity, as witnessed by the fact that small additions to our model can importantly increase the duration of the computation. However the tool is quite handy for building the model, and could be used to verify non-trivial properties of our system.

7. References

Alpcan T., Başar T., *Advances in Dynamic Games: Applications to Economics, Finance, Optimization, and Stochastic Control, Annals of Dynamic Games*, vol. 7, Birkhäuser, chapter Distributed Algorithms for Nash Equilibria of Flow Control Games, p. 473-498, 2005.

Alur R., Courcoubetis C., Dill D. L., “ Model-Checking in Dense Real-time”, *Information and Computation*, vol. 104, n° 1, p. 2-34, 1993.

Alur R., Dill D., “ Automata For Modeling Real-Time Systems”, *ICALP'90: Automata, Languages, and Programming*, vol. 443 of *Lecture Notes in Computer Science*, Springer, p. 322-335, 1990.

Alur R., Dill D., “ A Theory of Timed Automata”, *Theoretical Computer Science*, vol. 126, n° 2, p. 183-235, 1994.

Alur R., Feder T., Henzinger T. A., “ The benefits of relaxing punctuality”, *Journal of the ACM*, vol. 43, n° 1, p. 116-146, 1996.

Alur R., Henzinger T., Kupferman O., “ Alternating-time Temporal Logic”, *J. ACM*, vol. 49, n° 5, p. 672-713, 2002.

ANSI Std, T1.E1.4/2003-210R1, “ Very-high-bit-rate Digital Subscriber Lines (VDSL) Metallic Interface”, August, 2003.

Asarin E., Maler O., Pnueli A., Sifakis J., “ Controller Synthesis for Timed Automata”, *Proceedings of the 5th IFAC Cconference on System Structure and Control (SSC'98)*, Elsevier Science, p. 469-474, July, 1998.

Başar T., Olsder G. J., *Dynamic Noncooperative Game Theory*, SIAM, 1995.

Başar T., Srikant R., “ A Stackelberg network game with a large number of followers”, *J. Optimization Theory and Applications*, vol. 115, n° 3, p. 479-490, December, 2002.

Behrmann G., Cougnard A., David A., Fleury E., Larsen K. G., Lime D., “ UPPAAL-Tiga: Time for Playing Games!”, *in* W. Damm, H. Hermanns (eds), *Proceedings of the 19th International Conference, on Computer Aided Verification (CAV'07)*, vol. 4590 of *Lecture Notes in Computer Science*, Springer, p. 121-125, 2007.

Behrmann G., David A., Larsen K. G., “ A Tutorial on UPPAAL”, *SFM-RT'04*, vol. 3185 of *LNCS*, Springer, p. 200-236, 2004.

Bingham J. A. C., *ADSL, VDSL and Multicarrier modulation*, Wiley-Interscience, 2000.

Boyd S., Vandenberghe L., *Convex optimization*, Cambridge University Press, 2004.

Brihaye Th., Laroussinie F., Markey N., Oreiby G., " Timed Concurrent Game Structures", *in* L. Caires, V. T. Vasconcelos (eds), *Proceedings of the 18th International Conference on Concurrency Theory (CONCUR'07)*, vol. 4703 of *Lecture Notes in Computer Science*, Springer, Lisbon, Portugal, p. 445-459, September, 2007.

Chiang M., Tan C. W., Palomar D. P., O'Neill D., Julian D., " Power Control By Geometric Programming", *IEEE Transactions on Wireless Communications*, vol. 6, n° 7, p. 2640-2651, July, 2007.

Daws C., Olivero A., Tripakis S., Yovine S., " The tool KRONOS", *Hybrid Systems III: Verification and Control*, vol. 1066 of *Lecture Notes in Computer Science*, Springer-Verlag, p. 208-219, 1996.

Engwerda J. C., *LQ Dynamic Optimization and Differential Games*, Wiley, 2005.

Falomari D., Mandayam N., Goodman D., " A new framework for power control in wireless data networks: Games utility and pricing", *Proceedings of Allerton conference on communication, control and computing*, Illinois, USA, p. 546-555, September, 1998.

Fan X., T. A., Arcak M., Wenc T. J., Başar T., " A passivity approach to game-theoretic CDMA power control", *Automatica*, vol. 42, n° 11, p. 1837-1847, November, 2006.

Henzinger T., Prabhu V., " Timed alternating-time temporal logic", *FORMATS'06*, vol. 4202 of *LNCS*, Springer, p. 1-17, 2006.

İmer O. Ç., Başar T., *Constructive solutions to a decentralized team problem with application in flow control for networks*, Springer Verlag, chapter Topics in Control and Its Applications, p. 117-141, 1999.

Ji H., Huang C., " Non-cooperative uplink power control in cellular radio systems", *Wireless Networks*, vol. 4, n° 3, p. 233-240, 1998.

Krichen M., Tripakis S., " State Identification Problems for Timed Automata", *TestCom'05: Testing of Communicating Systems*, vol. 3502 of *Lecture Notes in Computer Science*, Springer, p. 175-191, 2005.

Larsen K. G., Pettersson P., Yi W., " UPPAAL: Status & Developments.", *CAV'97: Computer Aided Verification*, vol. 1254 of *Lecture Notes in Computer Science*, Springer, p. 456-459, 1997.

Lee J. W., " On Uniform Stabilization of Discrete-Time Linear Parameter-Varying Control Systems", *IEEE Transactions on Automatic Control*, vol. 51, n° 10, p. 1714-1721, 2006.

Long C., Zhang Q., Li B., Yang H., Guan X., " Non-Cooperative Power Control for Wireless Ad Hoc Networks with Repeated Games", *IEEE Journal on Selected Areas in Communications*, vol. 25, n° 6, p. 1101-1112, August, 2007.

MacKenzie A. B., Wicker S. B., " Game Theory and the Design of Self-Configuring, Adaptive Wireless Networks", *IEEE Communications Magazine*, November, 2001a.

MacKenzie A. B., Wicker S. B., " Game Theory in Communications: Motivation, Explanation and Application to Power Control", *GlobeCom2001*, San Antonio, Texas, p. 821-826, November, 2001b.

Maheswaran R. T., Başar T., “ Multi-user flow control as a Nash game: Performance of various algorithms”, *Proceedings of the 37th IEEE Conference on Decision and Control*, p. 1090-1095, December, 1998.

Maler O., Pnueli A., Sifakis J., “ On the Synthesis of Discrete Controllers for Timed Systems (An Extended Abstract)”, *in* E. W. Mayr, C. Puech (eds), *Proceedings of the 12th Symposium on Theoretical Aspects of Computer Science (STACS'95)*, vol. 900 of *Lecture Notes in Computer Science*, Springer-Verlag, p. 229-242, March, 1995.

Palomar D. P., Cioffi J. M., Lagunas M. A., “ Uniform Power Allocation in MIMO Channels: A Game-Theoretic Approach”, *IEEE Transactions on Information Theory*, vol. 49, n° 7, p. 1707-1727, July, 2003.

Palomar D. P., Fonollosa J., “ Practical Algorithms for a Family of Waterfilling Solutions”, *IEEE Transactions on Signal Processing*, vol. 53, n° 2, p. 686-695, February, 2005.

Palomar D. P., Lagunas M. A., Cioffi J. M., “ Optimum Linear Joint Transmit-Receive Processing for MIMO Channels with QoS Constraints.”, *IEEE Transactions on Signal Processing*, vol. 52, n° 5, p. 1179-1197, May, 2004.

Proakis J. G., *Digital Communications*, McGraw-Hill International Editions, 1995.

Shannon C. E., “ Communication in the presence of noise”, *Proc. Institute of Radio Engineers*, vol. 37, n° 1, p. 10-21, January, 1949.

Shum K. W., Leung K. K., Sung C. W., “ Convergence of Iterative Waterfilling Algorithm for Gaussian Interference Channels”, *IEEE Journal on Selected Areas in Communications*, vol. 25, n° 6, p. 1091-1100, 2007.

Subramanian A., Sayed A. H., “ Joint Rate and Power Control Algorithms for Wireless Networks”, *IEEE Transactions on Signal Processing*, vol. 53, n° 11, p. 4204-4214, November, 2005.

Yates R. D., “ A framework for uplink power control in cellular radio systems”, *IEEE Journal on Selected Areas in Communications*, vol. 13, n° 7, p. 1341-1347, 1995.

Yu W., Ginis G., Cioffi J., “ Distributed Multiuser Power Control for Digital Subscriber Lines”, *IEEE Journal on Selected Areas in Communications*, vol. 20, n° 5, p. 1105-1115, June, 2002.

Zander J., “ Performance of optimum transmitter power control in cellular radio systems”, *IEEE Transactions on Vehicular Technology*, vol. 41, n° 1, p. 57-62, 1992.