53rd IEEE Conference on Decision and Control
December 15-17, 2014. Los Angeles, California, USA

Formal methods for stability analysis of networked control systems with IEEE 802.15.4 protocol

Bo Wu, Hai Lin, Michael Lemmon

Abstract— **Wireless networked control systems (WNCS) with control loops closed over a wireless network are prevailing these days. However, due to uncertainties such as random accessing delays and possible packet drops, the stability analysis for WNCS is a challenging task. Most previous research on the communication network analysis either simply relied on Monte-Carlo simulation or followed the multi-state Markov chain framework. In this paper, we propose a probabilistic model checking methodology for stability analysis in which the communication system is modeled as a probabilistic timed automaton. The stability condition is written in the probabilistic temporal logic which can be checked and the satisfaction of the specification is equivalent to the stability of the WNCS. We then studied the impact of different MAC parameters on the satisfaction of the specification. Furthermore, if the specification is not satisfied initially, we propose a systematic way to design the controller so that the specification can be met. This work presents an attempt and a new angle to the communication and control system co-design problem.**

I. Introduction

Over the last decade, wireless networked control systems (WNCS) have enjoyed a great and increasing popularity in both academic and industrial world because of its flexible architectures, reduced installation and maintenance costs and distributed nature [1]. It finds applications in a wide range of areas such as mobile sensor networks [2], automated highway and unmanned aerial vehicles [3]. However it is also well known that multiple control loops being closed in the same network over the shared communication resources inevitably introduces new challenges in analyzing the closed loop system performance [4]. Indeed, WNCS suffers from random delays or packet drops introduced during transmission, channel access, retransmission and routing which may affect the control system stability or even destabilize the system [5].

This paper aims to investigate the stability issues of WNCS which involves modeling and analysis of both the control system and the communication network. Most research from controller design and analysis point of view usually skips the modeling of the communication network and simply assumes that its delay or packet drop properties are given [6], [7]. However, it may be difficult to obtain such properties in real applications, and these properties may be time varying.

On the other hand, there have been analytical studies of the communication protocols based on detailed multi-state Markov models, see e.g., [8]. The key approximation in [8] which is also adopted in most subsequent researches such as [9], [10] is the assumption of a constant and independent channel busy probability at each channel access attempt for each node, regardless of the number of backoffs or retransmissions already suffered. Again, this assumption may not always hold in practice.

Another direction of the stability analysis of WNCS is to first simulate the wireless network protocols and characterize their properties such as the delay distribution and so on, see e.g., [11] and [12]. However, in some cases, the simulation may be unrealistic and produce results that vary widely between different simulators and field experiments [13]. In fact, the impacts of simulator on the simulation results may be as much as the design of the protocol [14].

Motivated by these difficulties, we propose to use formal methods, especially model checking [15] to study the communication network and solve the stability analysis problem. Our basic idea is first to derive the stability condition from the control system and convert it into the specification that can be checked. Then we model the communication system as probabilistic timed automaton (PTA) and perform model checking to see if the specification is satisfied. If not, either we redesign the controller or tune the communication protocol parameters so that the specification can be met. Our idea of communication protocol tuning is inspired by the recent advances in communication area which targets for future communication networks with flexible and adaptive MAC layers [16].

The applications of formal methods for the analysis of the wireless communication protocol emerged recently from computer aided verification society and gained much interests. In [17] and [18], IEEE 802.11 protocol and IEEE 802.15.4 were modelled respectively using probabilistic timed automata (PTA) and the probabilistic model checking was performed using PRISM model checking tool [19]. Compared with the existing work, our novelty and main contributions can be summarized as follows. First, we convert the stability condition into probabilistic temporal logic that can be checked using model checking. Second, we model the communication network as PTA and study the impact of different MAC parameters on the satisfaction of specification. Third, given the model checking results and if the specification is not satisfied, we propose a controller design method to make sure that the specification can be met.

This paper is divided into nine parts. We give the problem formulation in the next section followed by a brief introduction on IEEE 802.15.4 protocol and probabilistic timed

Authors are with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, 46556 USA e-mail: (bwu3@nd.edu, hlin1@nd.edu, lemmon@nd.edu)

The financial supports from NSF-CNS-1239222 and NSF-EECS-1253488 for this work are greatly acknowledged.

automata in Section III and Section IV respectively. Section V provides the modelling of the communication network with PTA. In Section VI we provide the model checking results and analysis. In Section VII we propose how to redesign the controller to meet the specification. Section VIII shows an illustrative example. Section IX concludes the paper.

II. Problem formulation

Let's consider a WNCS with the star topology consisting of N linear time invariant (LTI) systems. The sensors are spatially distributed while the controllers and actuators are collocated in the center of the network. In this case, we do not have to consider the delay between the controller and the actuator and thus facilitate our analysis. The communication conforms to the IEEE 802.15.4 unslotted protocol which will be described in Section III. Assume that the i-th control system has the discrete time dynamic as follows:

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k) + C_i w_i(k) \tag{1}$$

where $x_i(k) \in \mathbb{R}^n, u_i(k) \in \mathbb{R}^m, w_i(k) \in \mathbb{R}^r$ are the state, control input and external input respectively. A_i, B_i, C_i are matrices of appropriate sizes and $u_i(k) = K_i d(k) x_i(k)$ is the control input where $K_i \in \mathbb{R}^{m\times n}$ and $d(k)$ is a Bernoulli random process. We assume that the full state of each plant is transmitted over the shared wireless network in the form of packets periodically. If a packet fails to completely arrive at the controller side within one sampling period, it will be discarded. For any k, $d(k)$ is 0 with probability p_i which indicates the packet is lost and $d(k)$ equals 1 with probability $1 - p_i$ meaning the packet is successfully received. The p_i is referred to as the packet dropping probability (PDP).

In many cases, the exact PDP may be very hard to determine. Furthermore, it has been noticed that when $n \geq 2$, even if the system is stable for some given PDP, it will not necessarily remain stable when the PDP is reduced indicating that the improvement of the network condition does not always result in an improved control system performance [20]. Therefore in [21], a quantity called the packet dropping margin (PDM) was introduced which is defined to be the maximum PDP that a WNCS can tolerate before becoming unstable in mean square sense. Formally [21],

Theorem 1. *For the i-th control system defined in (1), $\hat{A}_i = A_i + B_i K_i$, the system is mean square stable for any $p_i \leq PDM_i$. The PDM_i is given by*

$$PDM_i = \frac{1}{\mu(V_i)} \tag{2}$$

where

$$V_i = \begin{bmatrix} (S_i \otimes \hat{S}_i + \hat{S}_i \otimes S)(I - S_i \otimes S_i)^{-1} & \hat{S}_i \otimes \hat{S}_i \\ (I - S_i \otimes S_i)^{-1} & 0 \end{bmatrix}$$
$$S_i = \hat{A}_i \otimes \hat{A}_i, \hat{S}_i = A_i \otimes A_i - \hat{A}_i \otimes \hat{A}_i \tag{3}$$

and $\mu(.)$ is the largest positive eigenvalue of a matrix.

Therefore let $p = min\{PDM_1, ..., PDM_N\}$, from the definition of PDM, the specification to guarantee the mean square stability of all the N distributed control systems is that

- *The probability of all N stations successfully completing their transmissions within one sampling period is no less than $1 - p$.*

Given a WNCS, what we are interested in is first, whether the above specification is satisfied with the corresponding communication network and its protocol. Second, if not, how to tune the protocol parameters or redesign the controller to meet the specification. In Section V, we will model the communication network as a probabilistic timed automaton and in Section VI we will show how to transform the above specification into a time bounded probabilistic reachability checking problem. In Section VII, we will prove that if the specification is not satisfied, we can also redesign the controller so it can be met.

III. Overview of the unslotted IEEE 802.15.4

There are three types of nodes defined by the standard: coordinator, router and end devices. The coordinator serves to initiate a network and can allow others to join it. The routers are like the coordinators but they do not start a network on their own. The end devices only join the network. There must be and can only be one coordinator in each network but multiple routers and end devices. Every node gets access to the channel according to the unslotted CSMA/CA protocol. The protocol is implemented with units of time called backoff periods T_b which contains 20 symbol time T_s. One symbol has 4 bits and the bit rate is 250 $kbps$, so $T_s = 16\mu s$ and $T_b = 320\mu s$.

To send a packet, each node first initializes two variables : NB and BE. NB denotes the number of times that one node was required to backoff due to the busy channel. NB is initialized to 0 and upper bounded by NB_{max} whose default value is 4. BE, on the other hand, is the backoff exponent related to the maximum number of backoff periods before attempting to assess the channel. BE is initialized to the value of BE_{min} whose default value is 3 and cannot exceed BE_{max}, which is equal to 5 in the standard. No retransmission is assumed so ACK is not needed.

Whenever a node attempts to send a packet, it first initializes the variables and then uniformly delay a discrete time duration between $0T_b$ to $(2^{BE_{min}} - 1)T_b$. After it finishes waiting, it senses the channel to determine if the channel is occupied by other nodes or idle. Sensing channel typically takes T_b time. If the channel is busy, NB and BE will be automatically increased by 1 and the node goes into a second round of backoff by uniformly selecting a discrete time duration between $0T_b$ to $(2^{BE_{min}+1} - 1)T_b$. On the other hand, if the channel is sensed to be idle, the node will transmit the packet immediately. The collision may occur and packets will be lost if at least two nodes happen to sense the channel idle at the same time slot and start the transmission at the same time. As long as $NB \leq NB_{max}$, the node will return to backoff stage repeatedly when the channel is

sensed to be busy. When the number of backoff exceeds the maximum value, the packet will be dropped. We assume that the wireless channel is perfect, so the packets are lost only for two reasons: channel access failure or packet collision.

IV. Probabilistic timed automata

Probabilistic timed automata [22] are a modelling tool for distributed systems that involves time, nondeterminism, and probabilistic choices which is gaining popularity in recent years. Our notation follows [23].

A. Syntax

Let $\mathbb{T} \in \{\mathbb{R}^{\geq 0}, \mathbb{N}\}$ be the *time domain* of either the non-negative reals or naturals from which a finite set $\mathcal{X}$ of clocks take value. A function $v : \mathcal{X} \to \mathbb{T}$ is referred to as a *clock valuation*. For any $v : \mathcal{X} \to \mathbb{T}$ and $t \in \mathbb{T}$, the clock valuation $v \oplus t$ denotes the *time increment* for v with t.

Let $Zones(\mathcal{X})$ be the set of *clock constraints* over $\mathcal{X}$, which is a subset of the valuation space $\mathbb{T}^{|\mathcal{X}|}$ described by conjunctions of atomic constraints of the form $x \sim c$ for $x \in \mathcal{X}$, $\sim \in \{\leq, =, \geq\}$ and $c \in \mathbb{N}$. A clock valuation $v \in \mathbb{T}^{\mathcal{X}}$ satisfies a clock constraint ζ if and only if ζ resolves to true after substituting each clock $x \in \mathcal{X}$ with the corresponding value $v(x)$. We only consider the syntax of *closed, diagonal-free clock constraints*, in which atomic constraints of the form $x > c$ or $x < c$ (not closed) or $x - y \sim c$ (not diagonal-free) are not allowed.

A *discrete probability distribution* over a countable set Q is a function $\mu : Q \to [0, 1]$ such that $\sum_{q \in Q} \mu(q) = 1$.

Formally, a *probabilistic timed automaton* is a tuple $\mathcal{T} = (Loc, \bar{l}, \mathcal{X}, \Sigma, inv, prob)$ where:

- Loc is a finite set of *locations*;
- $\bar{l} \in Loc$ is the *initial location*;
- $\mathcal{X}$ is a finite set of *clocks*;
- Σ is a finite set of *events* disjoint from $\mathbb{T}$;
- $inv : Loc \to Zones(\mathcal{X})$ is the *invariant condition* function;
- $prob \subseteq Loc \times Zones(\mathcal{X}) \times \Sigma \times \mathsf{Dist}(2^{\mathcal{X}} \times Loc)$ is the *probabilistic transition relation*.

B. Semantics

There are two types of transitions that are chosen non-deterministically in each location of a probabilistic timed automaton: *Delay transitions* denotes the passage of time in the current location. They can be taken if the invariant condition is satisfied. *Event transitions* correspond to taking the probabilistic transitions $(l, g, \sigma, p) \in prob$. Namely, if the current location l satisfies the clock constraint g and the current event is σ, then $p((X', l'))$ denotes the probability of moving to the new location l' while resetting all clocks in X' to 0.

Formally, the semantics of probabilistic timed automata is given in terms of *timed probabilistic systems* [23]. The *dense-time semantics*, where $\mathbb{T} = \mathbb{R}$ and $\oplus = +$, is generally uncountable. The concept of a finite integral-time semantics [24] have been extended to PTA in [23]: for $\mathbb{T} = \mathbb{N}$ and $\oplus = \oplus_{\mathbb{N}}$, let $v \oplus_{\mathbb{N}} t = min\{v(x) + t, k_x + 1\}$. k_x is the largest value that the clock $x \in \mathcal{X}$ is compared to in all clock constraints of PTA. Because of the discrete time nature in the communication system, we will use the integral-time semantics which have been shown to preserve probabilistic reachability and expected reachability properties of closed, diagonal-free probabilistic timed automata [23].

V. Probabilistic timed automata models

In this section, we will present the probabilistic timed automata model for the communication network. The model is similar to [18] but we only focus on unslotted version without ACK and we do not consider urgent events.

A. Modelling assumptions

We consider a WNCS with the fixed star topology consisting of two linear time invariant (LTI) systems. The sensors are spatially distributed as end devices while the collocated controllers and actuators are the coordinator. Both systems start sending at the same time. The packet length in each system is nondeterministic within the same range and is integer times of T_b. We assume that there is no ACK sent by the receiver so there is no retransmission either. Each station has the same deadline which is their common sampling period h. So each packet has to be transmitted within time h or it will be discarded for the transmission of the newly sampled data. Furthermore, when backoff period has ended, the sender will perform a clear channel analysis (CCA) which we assume will take time $T_{CCA} = T_b$.

B. Modelling with PTA

In this study, we use PRISM to model the communication network as a PTA which is a parallel composition of the smaller modules namely the sender and the channel. Our modelling is quite generic with the protocol parameters that BE_{min}, BE_{max} and NB_{max} can all be modified to study their impact on the specification satisfaction.

Fig.1(a) illustrates the modelling of the first sender while the second sender is identical. x_1 is the clock. $BO_{PERIOD} = T_b$ and $data_1$ denotes the packet length. Fig.1(b) shows the channel model. c_1, c_2 denote the channel condition for sender 1 and 2 and initially they are both 0. When sending the packet, if there is no collision it can be seen that either $c_1 = 1$ or $c_2 = 1$. Otherwise $c_1 = c_2 = 2$ indicating that there is a collision. The overall system is the parallel composition of two stations and one channel module. After transmitting the packet, the system goes to the state "done" and stays there. As a result, this PTA models the transmission of one packet for each node which is enough for our Bernoulli jump linear system since the packet drop is an i.i.d process.

VI. Model checking and analysis

In this section, we will perform probabilistic model checking with the specification given in Section II and study the impacts of different protocol parameters on the satisfaction of the specification by the probabilistic model checker PRISM. With the model given in the previous section, we can readily

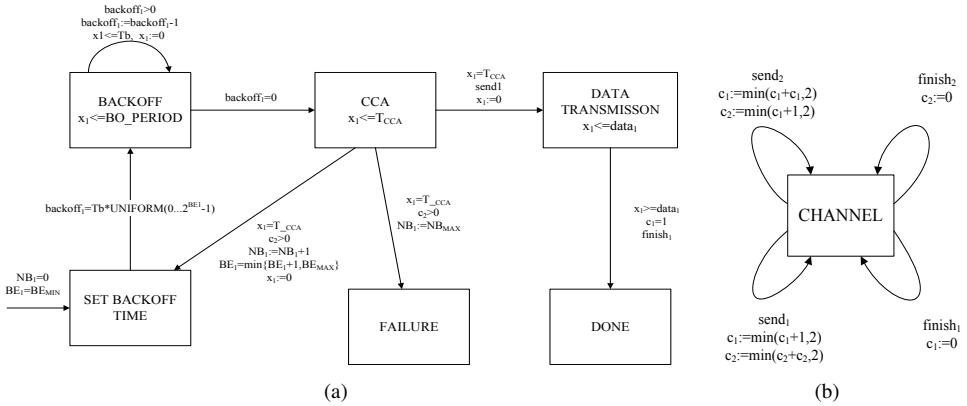


Fig. 1. Probabilistic timed automata and channel models for station in unslotted IEEE 802.15.4 CSMA/CA.

express the specification for guaranteeing stability into a bounded reachability formula in PTCTL (Probability Timed Computation Tree Logic) [22]. Due to the page limit, we will not explain the semantics of PTCTL here, interested readers may refer to [22]. As our models are PTAs, the probabilistic properties typically refer to the *minimum* or *maximum* probabilities over all possible scenarios.

To guarantee the stability specification from Section II, we need to check whether the following PTCTL formula is true:

$$z.(\mathcal{P}_{\geq 1-p}[true\ \mathcal{U}\ (done_1 \wedge done_2 \wedge z \leq h)])$$

where z is the *formula clock* denoting how long the communication system has run. $\mathcal{U}$ means *until* and $done_1,\ done_2$ are the states for two senders to reach. This PTCTL formula essentially carries the same meaning as the specification in Section II. For PTCTL model checking, actually we need to compute the probability over all possible scenarios and then compare with the probability bound to determine if a formula is satisfied. And the probability we are interested in is defined as follows:

- PR = *The minimum probability of both stations successfully transmitted their packets within one sampling period*

In the experiment, we set the data length to be nondeterministically varying between $20T_b$ to $30T_b$ for both stations and the sampling period to be $80T_b$. We are interested in how different parameters $BE_{min}, BE_{max}, NB_{max}$ affect PR. If $PR \geq 1-p$, then from section II we know that the overall WNCS is guaranteed to be m.s stable so we would like PR to be as high as possible.

Fig.2(a) shows the relationship between PR and BE_{min}. In this figure, $BE_{max} = 5, NB_{max} = 4$. It can be seen that the maximum PR can be achieved when $BE_{min} = 3$. It is because when BE_{min} is small, the backoff time will be relatively short and it is more likely that after backoff, the channel is still busy. On the other hand, if the BE_{min} is too large, even though the channel is less likely to be busy after backoff, the time spent in backoff will be larger and more likely to exceed the deadline.

Fig.2(b) indicates how PR changes with BE_{max}. In this figure, $BE_{min} = 3, NB_{max} = 4$. Similar tradeoff also exits in this situation.

Fig.2(c) represents how NB_{max} affects PR. In this figure, $BE_{min} = 3, BE_{max} = 5$. When NB_{max} is small, there is higher likelihood that the transmission fails due to exceeding the maximum number of backoffs. However it can be seen that when increasing NB_{max}, PR quickly levels off because even with higher NB_{max}, the transmission may still fail because the time exceeds the deadline. Note that we didn't pick larger NBs because it won't make too much sense to have too many number of backoffs given the time constraint to transmit the packet.

VII. Controller design for guaranteed system stability

In the previous section we check if the specification is satisfied. If so, we know that the control system is guaranteed to be stable. However, if the PR from the model checking cannot meet the control system requirement, in this section we will show that under certain conditions we can redesign the controller so that the specification can be met. To this regard, we will first need the following lemma from [21]:

Lemma 1. *If (1) is nominally stable, there holds*

$$PDM_i \geq \frac{1-\rho^2(\hat{A}_i)}{\kappa_2^2(\hat{A}_i)||A_i||_2^2 - \rho^2(\hat{A}_i)} \tag{4}$$

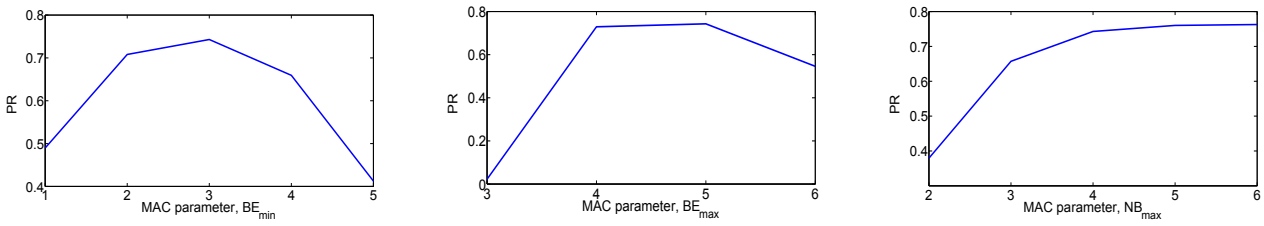
where $\hat{A}_i = A_i + B_iK_i$, $\rho(.)$, $||.||_2$ and $\kappa_2(.)$ denote the spectral radius, spectral norm and the spectral condition number of a matrix, respectively.

Then we have the following theorem:

Theorem 2. *For the i-th control system defined in (1), $\hat{A}_i = A_i + B_iK_i$, $PR = 1-\alpha, \alpha \in [0,1]$ from the probabilistic model checking, the system is mean square stablizable if the following holds:*

$$\alpha||A_i||_2^2 \leq 1 \tag{5}$$

and B_i has the right inverse $X_i \in \mathbb{R}^{m\times n}$ such that $B_iX_i = I_n$ where I_n denotes the identity matrix of dimension $n \times n$.



(a) (b) (c)

Fig. 2. PR with BE_{min},BE_{max} and NB_{max}.

Proof. From Lemma 1, we know that if we can design the controller K_i such that $\hat{A}_i = A_i + B_i K_i$ satisfies

$$\frac{1-\rho^2(\hat{A}_i)}{\kappa_2^2(\hat{A}_i)||A_i||_2^2 - \rho^2(\hat{A}_i)} \geq \alpha \tag{6}$$

then from (4) it is easy to find that $PDM_i \geq \alpha$. As a result, from the definition of PDM we know the system will be mean square stable. So the problem is now to design K_i such that (6) holds. From (6) we need to show

$$(1-\alpha)\rho^2(\hat{A}_i) \leq 1 - \alpha\kappa_2^2(\hat{A}_i)||A_i||_2^2 \tag{7}$$

when $\alpha = 1$, since $\kappa_2^2(\hat{A}_i) \geq 1$ by definition and $||A_i||^2 > 1$ meaning the open loop system is unstable, it can be found that (7) will not hold. So we can just focus on $\alpha \in [0, 1)$. Then we need to guarantee that

$$\rho^2(\hat{A}_i) \leq \frac{1 - \alpha\kappa_2^2(\hat{A}_i)||A_i||_2^2}{1-\alpha} \tag{8}$$

The nonnegativeness of left hand side of (8) requires its upper bound on the right side to be nonnegative as well which indicates the following

$$\alpha||A_i||_2^2 \leq \frac{1}{\kappa_2^2(\hat{A}_i)} \tag{9}$$

Then what is left is to find the $\hat{A}_i$ such that both (8) and (9) holds. In fact, there could be many such $\hat{A}_i$ among which we choose the diagonal matrix with positive and identical eigenvalues in the form of βI_n for simplicity in which $\beta \geq 0$. In this case, all its eigenvalues are the same and equal its singular values. Obviously $\kappa_2^2(\hat{A}_i) = 1$. Then from (8) and (9) we simply need

$$\begin{aligned} \beta^2 = \rho^2(\hat{A}_i) \leq \frac{1-\alpha||A_i||_2^2}{1-\alpha} \\ \alpha||A_i||_2^2 \leq 1 \end{aligned} \tag{10}$$

Then with the $\hat{A}_i$ we found, we can use the following equation to find K_i

$$K_i = X_i(\beta I_n - A_i) \tag{11}$$

Because

$$\begin{aligned} B_i K_i &= \beta I_n - A_i \\ A_i + B_i K_i &= \hat{A}_i = \beta I_n \end{aligned} \tag{12}$$

The above equalities result from the existence of the right inverse of B_i. The redesigned K_i will then guarantee the stability under given α. □

Remark 1. Theorem 2 clearly suggests a control and communication co-design framework. The key insight is from (5) which illustrates the trade-off between α and $||A_i||_2^2$. Note that α is the result from model checking indicating what quality of service in terms of the packet drop rate the communication system can offer given current network topology, protocol and parameters. On the other hand, $||A_i||_2^2$ from the control system's side has the inherent information on how stable and robust the open loop system is. Then in the circumstances when the specification is not satisfied initially, if we have a pair of α and $||A_i||_2^2$ such that (5) holds indicating either the packet drop rate is relatively small or the open loop system is not too unstable or both, it is then possible to design a controller such that the control system can tolerate such packet loss and remain stable. However if (5) does not hold, we then have to ask for better quality of service with smaller α from the communication system side with the guidance that's obtained from Section VI.

VIII. Illustrative Example

Consider a WNCS consisting of two plants in the form of (1) with

$$\begin{aligned}
A_1 &= \begin{bmatrix} -0.9240 & -0.2685 & 0.9993 \\ -1.2563 & 0.0528 & -0.5831 \\ -0.1972 & -0.3649 & 1.4306 \end{bmatrix} \\
B_1 &= \begin{bmatrix} -2.0071 & 0.7956 & -0.0311 & -0.9597 \\ 1.6595 & -0.7666 & -0.4059 & -0.3604 \\ -1.6422 & 1.5156 & 1.4775 & -0.8940 \end{bmatrix} \\
K_1 &= \begin{bmatrix} 0.9872 & 0.5563 & 0.3631 \\ 0.5553 & 1.2432 & -0.8672 \\ -0.5510 & -0.8739 & 0.1981 \\ -1.2832 & 0.1756 & -0.6510 \end{bmatrix} \\
A_2 &= \begin{bmatrix} 0.2844 & -0.9782 & 0.3754 \\ -0.1834 & 1.4031 & 0.1987 \\ 0.2226 & 0.3080 & -0.5885 \end{bmatrix} \\
B_2 &= \begin{bmatrix} 0.7784 & 0.2289 \\ -0.7940 & -0.2561 \\ -0.1330 & 0.0727 \end{bmatrix} \\
K_2 &= \begin{bmatrix} -0.4647 & 0.3285 & -0.5825 \\ -1.2156 & 1.4705 & -0.9587 \end{bmatrix}
\end{aligned} \tag{13}$$

From (2) and (3), it can be found that $PDM_1 = 0.2322$, $PDM_2 = 0.3265$. Then $p = min\{PDM_1, PDM_2\} = 0.2322$. Then the specification from Section II will be

- *The probability of both stations successfully completing their transmissions within one sampling period is no less than* $1 - p = 0.7678$.

Suppose we have the communication network described in Section V and Section VI with $BE_{min} = 2$, $BE_{max} = 5$ and $NB_{max} = 5$. From the model checking results in Fig. 4, we know that $PR = 0.7080$ in this case. Clearly $PR < 1-p = 0.7678$ and it is not possible to stabilize both systems. Observe that it is only node 1 that will be unable. So if we evaluate (5) for node 1, it will be

$$\alpha||A_i||_2^2 = 1.1067 \tag{14}$$

which is not smaller than 1. So to be able to redesign the controller, we need the communication system to tune its MAC parameters to provide better quality of service. If we increase BE_{min} to 3, from Fig. 4 we know $PR = 0.7429$. Even though in this case still we have $PR < 1-p = 0.7678$, evaluating (5) we get

$$\alpha||A_i||_2^2 = 0.9745 < 1 \tag{15}$$

B_1 has full rank so it has a right inverse. From Theorem 2, we know that there exists a K_1 such that the specification can be satisfied and both system will be stable. We can compute such K_1 by (10) and (11). Take $\beta = 0.15$, we can get

$$K_1 = \begin{bmatrix} 0.2216 & 0.0092 & 0.2640 \\ -0.1458 & 0.0537 & -0.2328 \\ -0.4608 & 0.0472 & -0.1855 \\ -1.6363 & -0.2560 & 0.3020 \end{bmatrix} \tag{16}$$

With this new controller K_1, it can be found that $PDM_1 = 0.3821$ and the specification PR is satisfied and thus the control system will be stable.

IX. Conclusion

In this paper, we studied the stability problem in WNCS which has been converted to a probabilistic model checking problem so that with formal methods we can have guaranteed performance. The PTA model has been built and the model checking has been performed on PRISM. The effect of different protocol parameters on reachability property has been analyzed. If the specification is not satisfied, then it is possible to redesign the controller to meet the specification. The model checking, communication parameter analysis and controller design together feature a control and communication co-design framework which involves trade-offs between control and communication systems.

References

[1] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, 2007.

[2] P. Ogren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *Automatic Control, IEEE Transactions on*, vol. 49, no. 8, pp. 1292–1302, 2004.

[3] P. Seiler and R. Sengupta, "An h approach to networked control," *Automatic Control, IEEE Transactions on*, vol. 50, no. 3, pp. 356–364, 2005.

[4] J. Baillieul and P. J. Antsaklis, "Control and communication challenges in networked real-time systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 9–28, 2007.

[5] M. B. Cloosterman, N. Van de Wouw, W. Heemels, and H. Nijmeijer, "Stability of networked control systems with uncertain time-varying delays," *Automatic Control, IEEE Transactions on*, vol. 54, no. 7, pp. 1575–1580, 2009.

[6] M. Lemmon and X. S. Hu, "Almost sure stability of networked control systems under exponentially bounded bursts of dropouts," in *Proceedings of the 14th international conference on Hybrid systems: computation and control*. ACM, 2011, pp. 301–310.

[7] N. W. Bauer, P. J. Maas, and W. Heemels, "Stability analysis of networked control systems: A sum of squares approach," *Automatica*, vol. 48, no. 8, pp. 1514–1524, 2012.

[8] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 3, pp. 535–547, 2000.

[9] P. Park, P. Di Marco, C. Fischione, and K. Johansson, "Delay distribution analysis of wireless personal area networks," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, 2012, pp. 5864–5869.

[10] E. Ziouva and T. Antonakopoulos, "CSMA/CA performance under high traffic conditions: throughput and delay analysis," *Computer Communications*, vol. 25, no. 3, pp. 313–321, 2002.

[11] J. Nilsson, "Real-time control systems with delays," Ph.D. dissertation, 1998.

[12] P. Park, P. Di Marco, P. Soldati, C. Fischione, and K. Johansson, "A generalized markov chain model for effective analysis of slotted IEEE 802.15.4," in *Mobile Adhoc and Sensor Systems, 2009. MASS '09. IEEE 6th International Conference on*, 2009, pp. 130–139.

[13] M. Fruth, "Formal methods for the analysis of wireless network protocols," Ph.D. dissertation, Oxford University, 2011.

[14] A. Fehnker and P. Gao, "Formal verification and simulation for performance analysis for probabilistic broadcast protocols," in *Ad-Hoc, Mobile, and Wireless Networks*. Springer, 2006, pp. 128–141.

[15] E. M. Clarke, O. Grumberg, and D. Peled, *Model checking*, 1999.

[16] I. Tinnirello, G. Bianchi, P. Gallo, D. Garlisi, F. Giuliano, and F. Gringoli, "Wireless mac processors: Programming mac protocols on commodity hardware," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 1269–1277.

[17] M. Kwiatkowska, G. Norman, and J. Sproston, *Probabilistic model checking of the IEEE 802.11 wireless local area network protocol*. Springer, 2002.

[18] M. Fruth, "Probabilistic model checking of contention resolution in the IEEE 802.15.4 low-rate wireless personal area network protocol," in *Leveraging Applications of Formal Methods, Verification and Validation, 2006. ISoLA 2006. Second International Symposium on*. IEEE, 2006, pp. 290–297.

[19] M. Kwiatkowska, G. Norman, and D. Parker, "Prism 4.0: Verification of probabilistic real-time systems," in *Computer Aided Verification*. Springer, 2011, pp. 585–591.

[20] S. Hu and W.-Y. Yan, "Stability of networked control systems under a multiple-packet transmission policy," *Automatic Control, IEEE Transactions on*, vol. 53, no. 7, pp. 1706–1711, 2008.

[21] ——, "Stability robustness of networked control systems with respect to packet loss," *Automatica*, vol. 43, no. 7, pp. 1243–1248, 2007.

[22] M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston, "Automatic verification of real-time systems with discrete probability distributions," *Theoretical Computer Science*, vol. 282, no. 1, pp. 101–150, 2002.

[23] M. Kwiatkowska, G. Norman, D. Parker, and J. Sproston, "Performance analysis of probabilistic timed automata using digital clocks," *Formal Methods in System Design*, vol. 29, no. 1, pp. 33–78, 2006.

[24] D. Beyer, "Improvements in bdd-based reachability analysis of timed automata," in *FME 2001: Formal Methods for Increasing Software Productivity*. Springer, 2001, pp. 318–343.