PTA as an approach to fault tree analysis

J.A. de Bie
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
j.a.debie@student.utwente.nl

ABSTRACT

This paper explores probabilistic timed automata (PTA) as an approach to fault tree analysis (FTA) by modelling two synthetic dynamic fault trees (DFTs) using both PTA and continuous timed Markov Chains (CTMC). Using UPPAAL, a simulation software, these models have been created and analysed with statistical model checking, in order to compare their availability, reliability and the time needed for verifying these systems. The availability is the percentage of time the system is operational within a given time period. The reliability is the probability that the system does not fail within a given time period. The results show that the PTA models yield the same availability as the CTMC models. However, the PTA models yield a lower reliability than CTMC. Furthermore, the PTA models require more time for the verification of the DFTs in this paper.

Keywords

System verification, fault tree (analysis), model checking, RAMS, PTA, CTMC

1. INTRODUCTION

To ensure the correct and safe working of safety-critical systems, they should satisfy the RAMS (Reliability, Availability, Maintainability, Security) requirement, which is often imposed by law. Safety-critical systems often concern important systems such as, air traffic control, nuclear power control and railroad infrastructure and are therefore important to society. Fault tree analysis (FTA) is a means to analyse the behaviour of systems and is a widely applied standard for RAMS analysis. There are many approaches to FTA, such as Binary Decision Diagram [4, 9] and Monte Carlo simulation [12]. The use of continuous timed Markov chains is a popular approach to FTA and uses exponential distributions to describe the system's behaviour. Another approach to FTA is the use of probabilistic timed automata (PTA), which can use a multitude of probability distributions including, binominal, uniform and exponential distributions and can therefore be used to model a multitude of systems, including communication protocols, aviation security systems, streaming download protocols and service level agreements [7]. This paper only analyses PTA using uniform distributions in comparison to CTMC by modelling two synthetic systems described by fault trees using both formalisms. Two Key Performance Indicators (KPI's) are analysed with these models. The first KPI is the availability, the second KPI is the reliability. According to definition 2, the availability is the percentage of time the system is in an operational state given a time interval. Definition 1 states that the reliability is the probability that the system does not fail within a given time period and is equal to $1 - F(t)$, where $F(t)$ is the cumulative probability distribution of the probability that the system fails within the given time period. Furthermore, the time used for the verification of the DFTs by the PTA and CTMC models are compared using UPPAAL. The goal of this paper is to investigate the usability of PTA in comparison to CTMC as an approach to FTA. To provide this comparison, the following questions will be investigated:

1. How does the simulated availability of the DFTs in figures 1 and 2 differ for the PTA and CTMC models?
2. How does the simulated reliability of the DFTs in figures 1 and 2 differ for the PTA and CTMC models?
3. How long does each formalism take to perform its verification of a system using UPPAAL?

Definition 1. The reliability at time t, $R(t)$, for a given time period T in $\mathbb{R} > 0$ can be retrieved from the probability of failure at time t, $F(t)$, of the system by the following equation:

$$Pr(t \leq T) = F(t) = 1 - R(t)$$

Definition 2. The availability at time t in $\mathbb{R} > 0$, $A(t)$, is the total time the system has been operational within a given time period and is described by the following equation:

$$A(t) = \frac{\text{Time operational}}{\text{Total time}}$$

In order to answer these questions, models will be created using UPPAAL for the dynamic fault trees (DFTs) [2] in figures 1 and 2 using both formalisms. The first DFT in Fig. 1 shows a system containing of two basic events (BEs), A and B, and an AND-gate. A BE is a representation of a physical component with a certain failure probability [2]. In this paper, the BEs that are considered can

28^{th} *Twente Student Conference on IT* February 2^{nd}, 2018, Enschede, The Netherlands.

also be repaired according to a certain repair probability. The first system fails if both BEs fail, but is operational as long as one BE is operational. The second DFT in Fig. 2 is more complex. It consists of three BEs, P, Q and S, two spare-gates, A and B, and an AND-gate. For spare-gate A, its primary component is P and for spare-gate B, its primary component is Q. Both A and B share a spare component S. The spare-gates fail when both their primary component has failed and its spare components have failed or are in use by other spare-gates. If both spare-gates fail, the system as a whole has failed. Using UPPAAL, the availability and reliability can be simulated over time. Furthermore, UPPAAL also provides the time it took to perform these simulations, which is used to answer question 3.

This paper is structured as follows. Section 2 provides more detail on the simulation tools that were used. Section 3 gives an introduction to DFTs and presents the UPPAAL models created to represent the DFTs figures 1 and 2 using PTA and CTMC. Section 4 presents the results of the availability, reliability and used verification time for each PTA and CTMC model. Section 5 presents an analysis of these results. Lastly, section 6 provides a conclusion of the research by answering the research questions and discusses future research.

There are several papers related to this research. A paper by Wu, Lemmon & Lin, 2017, explores PTA as an alternative modelling formalism to Monte Carlo simulation and multi-state Markov chains [11]. For the research, a PTA model was created for a network communication protocol. This model was then analysed using probabilistic model checking. With the main focus on the stability condition of the protocol. Similar to this paper, it explores PTA as an alternative to other modelling formalisms. However, it does not provide a comparison between different modelling formalisms, but focuses only on a PTA model. A paper by Norman, Parker & Zou, 2017, proposes an extension of PTA, namely, partially observable probabilistic systems (POPTAs) [8]. These allow local states to be partially visible to an observer or controller. Furthermore, automated techniques are presented for the verification of partially observable probabilistic systems. Partially observable Markov decision processes are (POMDPs) are used as well. In this paper POMDPs are used for discrete-time model and use POPTAs for dense time models. Experimental results are provided of POPTAs and POMDPs, which are analysed to compare aspects such as, the states generated and time used for each experiment. In the paper by Sproston, 2017, clock-dependent probabilistic timed automata (cdPTA) are introduced [10]. This variant on PTA uses clock bounds to determine when a transition can be made. The paper shows that the reachability problem [1, 3, 6] is undecidable for cdPTA with at least three clocks. The paper by Jurdziński, Laroussinie & Sproston, 2007, [5] already showed that this was undecidable for PTA using one or two clocks. The papers by Norma et al., Sproston and Jurdziński et al. are relevant to this research as these papers involve the analysis of PTA with clocks, which is also done in this paper. However, no comparisons of CTMC and PTA regarding availability, reliability and verification time have been found.

2. SIMULATION SOFTWARE

In order to simulate the availability and the reliability of the DFTs, UPPAAL SMC version 4.1.19 was used. UPPAAL was chosen because of its ability to analyse the behaviour of both CTMC and PTA models. Furthermore, it provides an easy-to-use GUI for creating these models.

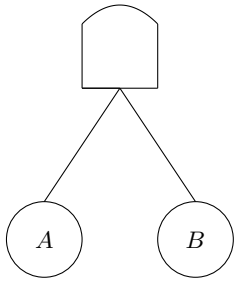

Figure 1. DFT with an AND gate and two BE-components; A and B.

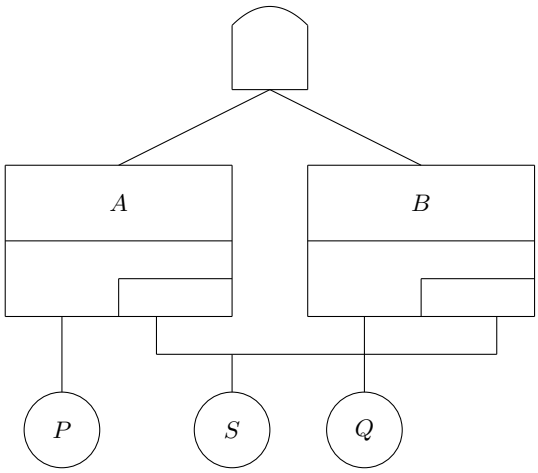

Figure 2. DFT with an AND gate, spare-gates; A and B, and three BE-components; P, Q and S.

The hardware used for performing the simulations using UPPAAL was a 6-core, 3.50-3.70GHz Intel® Core™ i7-5930K, with 12GiB 2400MHz of available DDR4 RAM. The importance of using the same program to simulate the CTMC and PTA models comes from the intention to compare the simulations of their availability, reliability and verification time. Due to the empiric nature of research question 3, it is important to use the same hardware as well. By using the same environment for each simulation, influences due to unwanted factors are limited. This means that each simulation is more consistent and allows for a more accurate analysis of the results.

3. DFT MODELS

Fault trees (FT) are a means to model failures of a system's components using static logic gates and basic events (BEs). This allows for analysis of the behaviour and interactions between components of a system using Boolean logic. DFTs extend fault trees by introducing dynamic logic gates, such as the spare, PAND and FDEP gates [2]. These additional gates allow DFTs to be used for the analysis of complexer systems than is possible with regular fault trees. At the leaves of fault trees, BEs are located. To give an example of a BE, consider a light bulb. The light bulb can fail due to its filament burning through. it can also be repaired by replacing it with a new light bulb. Regarding the dynamic gates introduced by DFTs: This paper only explores an DFT which uses spare-gates, whose behaviour is described in section 3.5. The DFTs in this paper are not based on real systems and are not built to be realistic, but rather to explore the results of the PTA and CTMC models in comparison to each other. The parameters of these DFTs are arbitrarily chosen.

3.1 Modelling the BE-component

Using UPPAAL, the DFTs were translated into PTA and CTMC models. The model in Fig. 3 represents a BE component for the CTMC model. Fig. 4 shows the model for the BE component for the PTA model. The BEs that are considered in this paper can either fail or be repaired. Therefore, they can be described using two states:

OP	This component is operational.
FAIL	This component is nonoperational.

The CTMC model of the BE has two parameters:

λ_{CTMC}	The exponential failure rate of the BE.
μ_{CTMC}	The exponential repair rate of the BE.

The exponential failure rate, λ_{CTMC}, determines the time the system can spend in the OP state before the transition from OP to FAIL will be made. Similarly, the exponential repair rate determines the time the system can spend in FAIL before it must take the transition from FAIL to OP. For the PTA model of the BE, the same states are used as for the CTMC model. However, the PTA model has four parameters:

λ_{PTA}	The mean time to failure (MTTF) of the BE.
μ_{PTA}	The mean time till repair (MTTR) of the BE.
δ_f	The variation in time around the MTTF. Used to determine the lower and upper clock bounds for the PTA.
δ_r	The variation in time around the MTTR. Used to determine the lower and upper clock bounds for the PTA.

One parameter that is used by both the CTMC and PTA models is *id*. This parameter is a unique number assigned to each instance of the PTA or CTMC models and can be used to establish channels between models. The channels F and R are the last parameters found in both the PTA and CTMC models. However, these will be discussed in section 3.2. The last variable, c, is not a parameter, but represents the clock attached to each instance of the PTA model. This clock starts running when the simulation of the model starts. The PTA model uses the clock in conjunction with *guards* and *invariants*. Guards determine the prerequisite for taking a transition. In Fig. 4 the guards are located above the transitions. The guard above the transition from OP to FAIL determines that the transition can only be taken when the clock, c, is greater or equal to $\lambda_{PTA} - \delta_f$. Invariants, on the other hand, determine how long the system is allowed to reside in a state before it must take a transition. The invariants can be seen in Fig. 4, next to the states. The invariant next to OP determines that the system can reside in OP as long as clock c is smaller or equal to $\lambda_{PTA} + \delta_f$. The statement $c = 0$ means that clock c is reset to zero when the transition is taken. The probability that a transition is taken within the bounds imposed by the invariants and guards is determined by a uniform distribution with these bounds. Therefore, the clock can be used together with the guards and invariants, to determine the repair and failure rates according to an uniform distribution with a lower and upper bound. In the PTA model of the BE, the failure rate is determined by a uniform distribution with lower bound

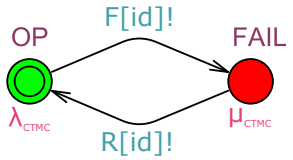

Figure 3. UPPAAL CTMC model template of the BE component

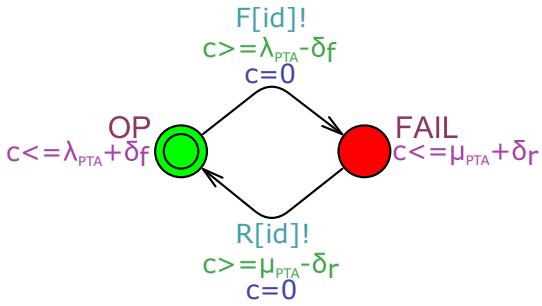

Figure 4. UPPAAL PTA model template of the BE component

$\lambda_{PTA} - \delta_f$ and upper bound $\lambda_{PTA} + \delta_f$. This means that on average, the transition from OP to FAIL will be taken when c equals λ_{PTA}. Using the same technique, the repair rate is set to be on average μ_{PTA}.

3.2 Modelling the AND-gate

The next DFT component that was modelled is the AND-gate. The AND-gate fails when all components attached to it have failed. For the DFTs that are considered in this paper, the AND-gate has two components attached to it. To describe the AND-gate, the model in Fig. 5 was created. In addition to the states OP and FAIL, it also has the following states:

FAIL_A	The component defined by id, A, has failed.
FAIL_B	The component defined by id, B, has failed.

Furthermore the model of the AND-gate also uses the following parameters:

F	The channels, each defined by an id, which are used to communicate the failure of the component A, to another model template.
R	The channels, each defined by an id, which are used to communicate the repair of the component B, to another model template.

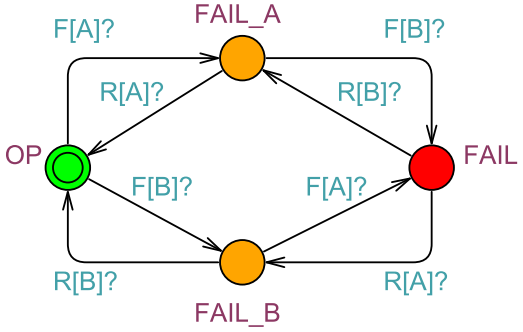

Figure 5. UPPAAL model template of the AND-gate component

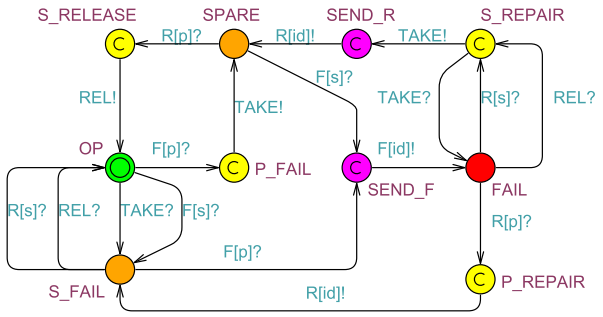

Figure 6. UPPAAL model template of the spare-gate component

As mentioned in the previous section, each instance of a model is provided with a unique integer to identify it. This is mainly used to create channels between different models. Channels are communication lines between models which can be used to send signals from one model to another. A signal is sent when a transition is taken and the synchronization statement is followed by a exclamation mark. The question mark denotes that the transition can only be taken when a signal is sent over that channel. This allows models to take a transition when a transition by another model is taken. For example, assume the model of the AND-gate is initialized with the id's of BEs A and B. Then when BE A fails, it sends a signal over channel $F[A]$. This signal will be received by the AND-gate as it is initially in the OP state. As a result of the signal over channel $F[A]$, the AND-gate moves from OP to FAIL_A. In order to analyse the models, UPPAAL required all channels to be broadcast. Therefore, all channels used in this paper are broadcast channels. This means that when a signal is sent over such a channel, every model that is listening for it will take the transition. That is, if the invariants and guards are satisfied.

3.3 Modelling the spare-gate

The last component which is modelled in UPPAAL is the spare-gate. The spare-gate has a primary component and spare components attached to it. The spare-gate is operational as long as the primary component or a spare component is operational and available. If the primary component fails, the spare-gate will switch to its spare-component, if there is one available. If the spare component is shared, it might be operational but unavailable as it is in use by another spare-gate. When both the primary and secondary components have failed, then the spare-gate has failed as well. The model of the spare-gate in Fig. 6 only considers one primary component for each spare-gate and one spare component which is shared with a second spare-gate. In this case, the spare component is a hot-spare. Meaning that it is subject to failure and repair, regardless of whether it is being used by a spare-gate. As opposed to a cold-spare, which can only fail when it is being used. As a result, the spare-component fails with the same probability as the primary component. The model of the spare-gate has the following states in addition to the OP and FAIL state:

SPARE	● This spare-gate has taken control of the spare component and its primary component has failed.
S_FAIL	● The spare component has failed or is in use by the other spare-gate.
P_FAIL	● The primary component has failed, but the spare component is available.
P_REPAIR	● The primary component has been repaired after both the primary and spare component had failed.
S_REPAIR	● The spare component has been repaired after both the primary and spare component had failed.
S_RELEASE	● The primary component has been repaired, but the spare-gate still has control of the spare component.
SEND_F	● Both the primary and spare component have failed, but the spare-gate has not sent the signal that it is non-operational.
SEND_R	● The spare-gate has taken control of the spare component, but has not sent the signal that it is operational.

Furthermore, the following parameters are introduced:

P	The id of the primary component of this spare-gate
S	The id of the spare component of this spare-gate
TAKE	The channel that is used to communicate which spare-gate takes control of the spare component.
REL	The channel which is used to communicate which spare-gate releases control of the spare component.

The channels $F[P]$ and $F[S]$ are used to communicate the failure of the primary and spare component respectively. Channels $R[P]$ and $R[S]$ serve the same purpose, but communicate the repair of the primary and spare component. The failure and repair of the spare-gate are communicated through channels $F[id]$ and $R[id]$, where id is the unique identifier for the spare-gate. The model also introduces a new type of state, denoted with a c. These states are referred to as committed states. When the spare-gate is in such a state no time passes, and a transition must be taken before all other possible transitions in the system. In the model of the spare-gate, these states are used to immediately send signals in response to received signals. For example, when the spare-gate is in the FAIL state and its primary component is repaired, it receives a signal over channel $R[P]$ and moves to P_REPAIR accordingly. P_REPAIR is chosen to be a committed state, such that the transition to S_FAIL is immediately made. This means that the repair of the spare-gate over channel $R[id]$ is immediately communicated to other components. Committed states allow other components that depend on the spare-gate to be immediately notified when the spare-gate is repaired or has failed. Lastly, the model of the spare-gate has two new channels; TAKE and REL. TAKE allows two spare-gates to communicate which spare-gate is using the spare component. REL is used to communicate the release of the spare component by a spare-gate.

3.4 Basic DFT: AND-gate with repairs

The first DFT can be found in Fig. 1. It consists of two BEs and an AND-gate. BEs A and B can fail independently, but because of the AND-gate, the complete system fails when both A and B fail.

Two instances of the BE model in Fig. 3 or Fig. 4 will be made, depending on whether a CTMC or PTA model is created. Both instances have an own unique id which will be called A and B for convenience. To complete the system, one instance of the AND-gate model in Fig. 5 is created with the id's of the BE models as its arguments. When, BE A fails, it takes a transition from OP to FAIL and will send a signal over channel $F[A]$. In response, the AND-gate model makes a transition from OP to FAIL_A. If BE B fails afterwards, a signal is sent over channel $F[B]$ and both BEs will be in the FAIL state. This causes the AND-gate to move from the FAIL_A state to the FAIL state, meaning that the whole system has failed. If BE A is repaired, the BE moves from the FAIL state to the OP state and sends a signal over the repair channel, $R[A]$. This will have as result that the AND-gate moves from the FAIL state to FAIL_B. The failure and repair rate determine the time the system remains within a state before jumping from that state to another. For the CTMC models this is determined by the exponential distributions with rates λ_{CTMC} and μ_{CTMC} for the failure and repair rate respectively. In case of the PTA models, the failure and repair rate are each determined by a uniform distribution with lower bounds $\lambda_{PTA} - \delta f$ and $\mu_{PTA} - \delta r$ and upper bounds $\lambda_{PTA} + \delta_f$ and $\mu_{PTA} + \delta_r$ respectively.

3.5 DFT with spare-gates

The second DFT that has been analysed can be found in Fig. 2. It consists of an AND-gate, with two spare-gates, A and B, attached to it. Spare-gate A uses BE P as its primary component. Spare-gate B has BE Q as its primary component. Both spare-gates share BE S as their spare component. This system fails when both spare-gates fail due to the behaviour of the AND-gate. A spare-gate fails when both its primary and spare component fails. A detected failure of the spare component by a spare-gate can have one of two causes. Either the spare component has become non-operational, or it is operational, but in use by the other spare-gate.

Depending on the CTMC and PTA model, three instances of the model in Fig. 3 or 4 are created to represent BEs P, Q and S. Furthermore, two instances of the model in Fig. 6 are created to represent spare-gates A and B. To complete the system, one instance of the model in Fig. 5 is created to represent the AND-gate. To better explain the working of these models, consider the following scenario: When the spare BE, S, fails, a signal is sent over channel $F[S]$. In response, both spare-gates will move from OP to S_FAIL. If then BE Q fails, a signal is sent over channel $F[Q]$ and spare-gate B moves from S_FAIL to SEND_F as its primary component has failed. SEND_F is a committed state, meaning that in this state no time passes, and that it must take a transition to another state before every other possible transition. This means that upon arriving in SEND_F, immediately the transition is taken to FAIL and a signal is sent to the AND-gate over channel $F[B]$. In response the AND-gate moves from OP to FAIL_B as spare-gate B has failed. If then the spare component is repaired, a signal is sent over channel $R[S]$. As a result, spare-gate A will move from S_FAIL to OP and spare-gate B will move from FAIL to S_REPAIR, another committed state. The next transition made by spare-gate B will be from S_REPAIR to SEND_R and a signal is sent over channel TAKE. This will cause spare-gate A to move from OP back to S_FAIL. SEND_R is also a committed state and therefore the next transition is again made by spare-gate B. The spare-gate moves from SEND_R to SPARE, which denotes that it is using the spare component. The signal that is send over $R[B]$ by taking this transition will cause the AND-gate to return from FAIL_B to OP. If BE Q is repaired, a signal is sent over $R[Q]$ and spare-gate B will move from SPARE to S_RELEASE. Meaning that the spare-gate will switch from using the spare component to its primary component. The next transition is immediately taken from S_RELEASE to OP, causing a signal to be sent over channel REL. In response, spare-gate A moves from S_FAIL to OP and both spare-gates are fully operational again.

Algorithm 1 for measuring the availability

```
 1: clock c
 2: double T_f = 0.0
 3: double T_o = 0.0
 4: double a = 1.0
 5:
 6: procedure addFailureTime
 7:     T_f = T_f + c
 8:     c = 0
 9:     calculateAvailability
10: end procedure
11:
12: procedure addOperationalTime
13:     T_o = T_o + c
14:     c = 0
15:     calculateAvailability
16: end procedure
17:
18: procedure calculateAvailability
19:     if T_f + T_o > 0 then
20:         a = T_o/(T_o + T_f)
21:     end if
22: end procedure
```

3.6 Script

In order to create a graph of the availability of each system, a script was created. The pseudocode of this script can be found in algorithm 1, and is used in conjunction with the AND-gate model. The script consists of three functions and contains 4 variables. The first variable, c, is the clock unique to the instance of the AND-gate model. This is used to measure the time the model spends in each state during a simulation. The variables T_f and T_o contain the sums of time that the model has spent in a failure state or an operational state. The final variable a is used to store the availability of the model. Whenever a transition is made from a failure state to an operational state, the procedure *addFailureTime* is executed. The other procedure, *addOperationalTime*, is executed whenever a transition is taken from an operational state to another operational state and whenever a transition is taken from an operational state to a failure state. The execution of the procedures, whenever a transition is made, is necessary to make sure the availability, a, gets updated regularly during the simulation. This yields less erratic transient plots of the availability when running simulations and increases the accuracy of the simulated availability. To prevent a division by zero in line 20 of the algorithm, a check is made to ensure the denominator, $T_o + T_f$, is greater than zero. For this scenario to take place, the clock, c, has to

be zero when the first transition is made. This will cause both T_o and T_f to be zero when *calculateAvailability* is called. The possibility of this scenario is very slim, as the chance that a transition will be made from one state to another, immediately when the simulation starts, is very small. By plotting the availability, a, during a simulation, figures, such as Fig. 7, can be created and analysed.

4. RESULTS

This section presents the results of the simulations of the availability and reliability and the verification time of two systems described by the DFTs from figures 1 and 2. From the models from section 3, the PTA and CTMC models were created for these DFTs. To simulate these systems, values had to be chosen for the failure rates and repair rates of their BEs. In order to determine these rates, values for the MTTF and MTTR were chosen. The MTTF is the average time a system or component takes to move from its operational state, to its failure state. For the BE models in figures 3 and 4 of this paper, this means the average time the BE spends in the OP state before moving from OP to FAIL. The MTTR is the average time it takes to repair a component or system. For the BE models this is the average time a BE spends in the FAIL state before moving to the OP state. Using definitions 3 and 4, the MTTF and MTTR can be used to calculate the exponential repair and failure rate needed for the CTMC models. Furthermore, these values are used to determine the clock bounds for the PTA models.

4.1 Parameters

For all simulations, the BEs were given a MTTF and MTTR of 14 days and 2 days respectively. Using definitions 3 and 4 it can be determined that the BEs of the CTMC models have an exponential failure rate, λ_{CTMC}, of 1/14 and an exponential repair rate, μ_{CTMC}, of 1/2. For the BEs of the PTA models, λ_{PTA} and μ_{PTA} are 14 and 2 days respectively and δ_f and δ_r are 7 days and 1 day respectively. The simulations of the reliability and availability were done using a probability uncertainty of 0.0001.

Definition 3. The relation between the exponential failure rate, λ, and the $MTTF$, the mean time with which the system or component moves from its operational state to its failure state, can be described by the following equation:

$$1/MTTF = \lambda$$

Definition 4. The relation between the exponential repair rate, μ, and the $MTTR$, the mean time it takes to repair a component or system, can be described by the following equation:

$$1/MTTR = \mu$$

4.2 Availability

The script from section 3.6 was used to create the simulations of the availability. During the simulation of the system, the availability is updated whenever a transition is made by the AND-gate. Figures 7 and 9 show the moving average of the availability of both the PTA and CTMC models of the basic DFT in Fig. 1. For the DFT with spare-gates in Fig. 2, the moving average of the availability for the PTA and CTMC models is shown in figures

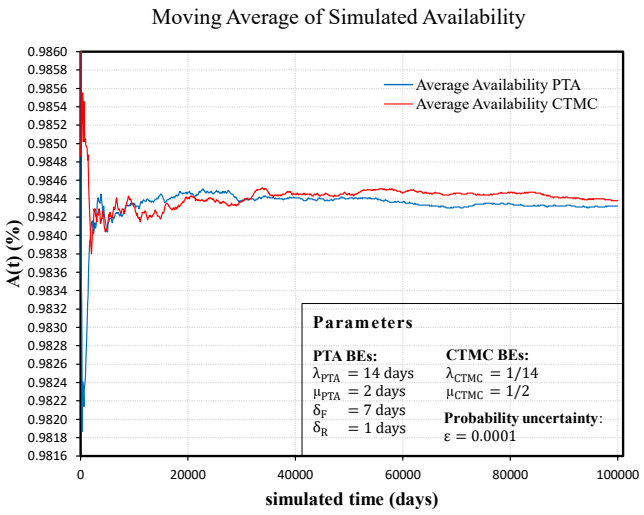

Figure 7. The average availability of the CTMC and PTA model of the DFT in Fig. 1.

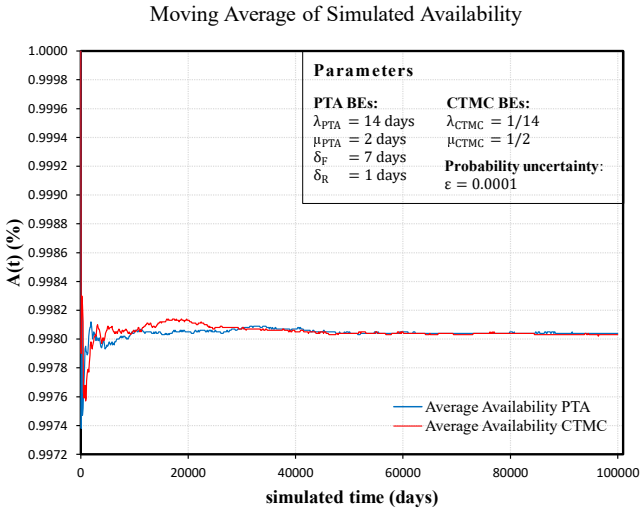

Figure 8. The average availability of the CTMC and PTA model of the DFT in Fig. 2.

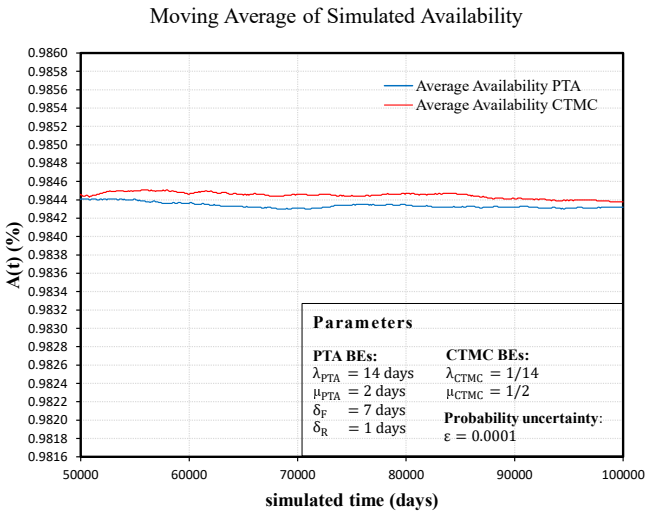

Figure 9. The steady-state behaviour of the availability of the CTMC and PTA model of the DFT in Fig. 1.

8 and 10. The moving averages were calculated by performing 50 simulations of 100,000 days of the availability, with a probability uncertainty of 0.0001. Using UPPAAL, 50 sets of coordinates of the simulated availability are retrieved for each model. Not all of these sets have the same dimensions. Therefore, in order to calculate the mean of these coordinates, each set had to be interpolated along a common time-axis. By using linear interpolation, the coordinates were calculated such that all sets had the same dimensions. From the interpolated coordinates the average simulated availability was calculated. For the basic DFT, the minimum and maximum availability retrieved from the 50 simulations are approximately 98,31% and 98,57% for the CTMC model and 98,34% and 98,53% for the PTA model. For the DFT with the spare-gates, the maximum and minimum availability are approximately 99,75% and 99,84% for the CTMC model and 99,76% and 99,84% for the PTA model. Figures 7 and 8 show that it takes about 50,000 days for the availability to reach a steady-state. Differences between the sum of the operational time and the sum of the non-operational time becomes more consistent as more time passes. Using definition 2, it can be concluded that the availability will also become more consistent over time. Figures 9 and 10 provide a more detailed view of the steady-state behaviour of the average availability of the DFTs. The graphs clearly show that both the CTMC and PTA models yield approximately the same availability for both DFTs. The availability of the PTA model in Fig. 9 is lower than the CTMC model. However, this difference of is so small it is considered negligible.

4.3 Reliability

In order to determine the reliability of each DFT, the probability of failure, $F(t)$, has to be known. According to definition 1, $F(t)$ is a cumulative probability distribution. To be more exact, $F(t)$ is the cumulative probability that the system moves from its initial operational state to a non-operational state within a given time interval. Using UPPAAL's verifier, the queries Pr [<=1000] (<> ANDGATE.FAIL)] and Pr [<=10000] (<> ANDGATE. FAIL)] were created. These queries check the probability that within, either a 1000 or 10,000 days, the AND-gate model template moves to the FAIL state. In other words, it checks the probability of failure of a system within a given time period, $F(t)$. The first query of 1000 days was used for simulating $F(t)$ for the basic DFT. The second query of 10,000 days was used for the DFT with spare-gates. For the DFT with spare-gates it was necessary to increase the time-period, because $F(t)$ increased slower than for the first DFT. Using definition 1, these results were used to calculate the reliability of each PTA and CTMC model. For the CTMC and PTA models, the reliability has been plotted in figures 11 and 12. For all simulations of the reliability, the probability uncertainty was set to 0.0001.

4.4 Simulation time

To measure differences in the required time for the simulations, the query Pr[<=200](<> ANDGATE.FAIL) was created in UPPAAL. This query calculates the probability that the AND-gate model moves from its initial operational state to its non-operational state within 200 days. The time period of 200 days was chosen, because UPPAAL required more computation time to perform this query as opposed to the same query, but with a time period of 1000 days. Furthermore, it can be seen in figures 11 and 12 that the PTA models are simulated for a shorter time. To prevent this causing a difference in the time used to verify the query, the time period was set to be 200 days. The

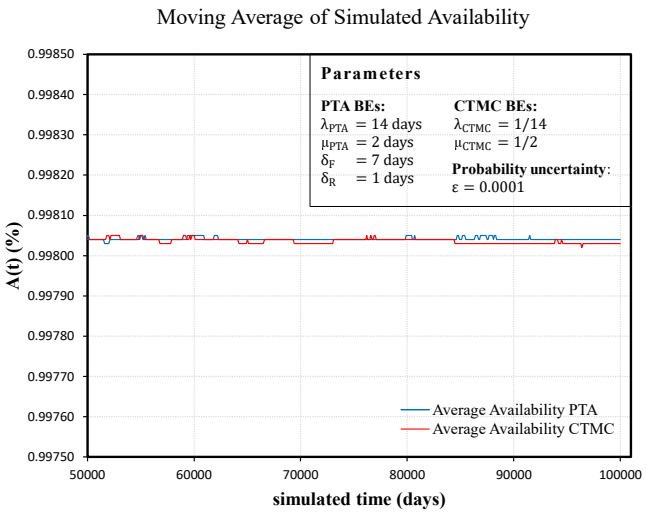

Figure 10. The steady-state behaviour of the availability of the CTMC and PTA model of the DFT in Fig. 2.

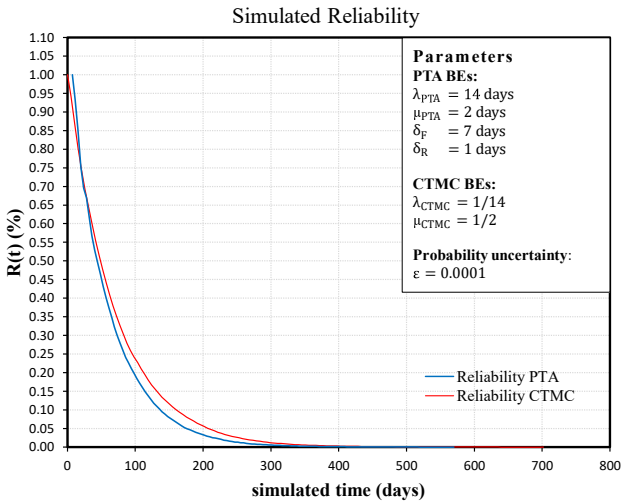

Figure 11. The reliability of the CTMC and PTA model of the DFT in Fig. 1.

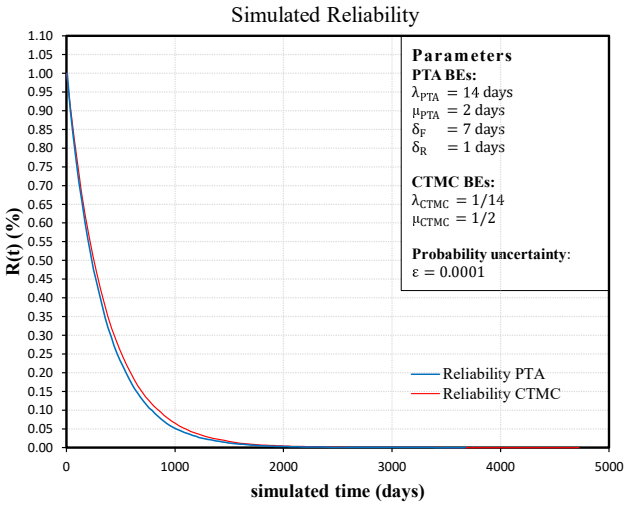

Figure 12. The reliability of the CTMC and PTA model of the DFT in Fig. 2.

query was run with a probability uncertainty, ϵ, of 0.001 for the basic DFT in Fig. 1 and the DFT with spare-gates in Fig. 2. The number of runs of the query, the total time used by each model to perform these runs and the average time used per run are presented in tables 1 and 2. From these results it can be deduced that the verification of the PTA model was on average 0.006 milliseconds slower per run for the basic DFT, and 0.067 milliseconds slower per run when considering the DFT with spare-gates.

Table 1. Time used for verifying Pr[<=200](<> ANDGATE.FAIL) for the DFT from Fig. 1 in seconds. $\epsilon = 0.001$

Model	Runs	Tot. time (s)	time/run (ms)
CTMC	1009867	59.510	0.059
PTA	644760	41.983	0.065

Table 2. Time used for verifying Pr[<=200](<> ANDGATE.FAIL) for the DFT from Fig. 2 in seconds. $\epsilon = 0.001$

Model	Runs	Tot. time (s)	time/run (ms)
CTMC	4675385	1027.50	0.219
PTA	4750630	1360.893	0.286

To further investigate the time used for simulations by each model, another query was created. Namely, simulate 100 [<=1000000](ANDGATE.availability), which performs 100 simulations of the availability of the system. For each PTA and CTMC model, this query was performed 5 times with a probability uncertainty of 0.0001, bringing the number of simulations performed by each model to 500. The total time required to perform these 500 simulations and the average time used per simulation for the PTA and CTMC models are presented in table 3 for the basic DFT and table 4 for the DFT with spare-gates. For the basic DFT, the PTA model is approximately 0.239s slower per simulation than the CTMC model. For the DFT with spare-gates, the PTA model is approximately 0.675s slower per simulation than the CTMC model.

Table 3. Time used for verifying simulate 100 [<=1000000](ANDGATE.availability) for the DFT from Fig. 1 in seconds. $\epsilon = 0.0001$

Model	Number of Simulations	Total time (s)	time /simulation (s)
CTMC	500	279.656	0.559
PTA	500	399.126	0.798

Table 4. Time used for verifying simulate 100 [<=1000000](ANDGATE.availability) for Fig. 2 in seconds. $\epsilon = 0.0001$

Model	Number of Simulations	Total time (s)	time /simulation (s)
CTMC	500	793.393	1.587
PTA	500	1130.969	2.262

5. DISCUSSION

In this section the process of creating the models is discussed. Furthermore, the results are analysed and future work is discussed.

5.1 Creating the models

Creating the models for the BEs and AND-gate was quite simple. The BE models were not subject to a lot of changes throughout the research process. Creating the model for the spare-gates proved to be more difficult. The model found in Fig. 6 was created after several revisions and has been simplified several times. However, there were no significant differences in creating the models for the CTMC and PTA. The PTA model of the BE used two additional parameters and a clock, but this did not affect the difficulty to work with PTA.

5.2 Key Performance Indicators

When analysing the availability of both DFTs, the erratic transient behaviour at the start of the simulation shown in figures 7 and 8 is disregarded. This behaviour is due to the script from section 3.6, used to measure the availability. Instead the steady-state values shown in figures 9 and 10 are analysed. From these figures it is clear that both formalisms give approximately the same result. To calculate the average availability, 50 simulations for each model were performed. In the case of the first DFT, these 50 simulations yielded an availability between 98.31% and 98.57% for the CTMC model, and between 98.34% and 98.53% for the PTA model. For the second DFT, the simulations showed that the CTMC model had an availability between 99,84% and 99,75%. For the PTA model this lies between 99,84% and 99,76%. Figures 9 and 10 show that the average availability is approximately the same for both the PTA and CTMC models. The differences between the models can be considered negligible, due to the slight difference between each simulation that was performed. Two simulations of the same system will never be exactly the same due to the use of probability distributions by both the CTMC and PTA. From these simulations it is clear that both modelling formalisms can be used to approximate the availability of the systems considered in this paper. However, for more complex systems it has not yet been proven that both formalisms can be used and could be an interesting extension of this research.

The reliability of both DFTs in figures 11 and 12 show some differences between the PTA models and CTMC models. The first difference is that the simulation of the PTA models lasted shorter than those of the CTMC models. This can be explained due to the fact that PTA models use uniform distributions which have a lower and upper bound. Outside these bounds, the probability is always 0. A transition in a model will have to be taken within these bounds. On the other hand, the CTMC models use exponential distributions, which have only a lower bound of 0, but no upper bound. The probability approximates 0 as time passes, but it never reaches it. This means that in a CTMC model, a transition could take infinitely long, though the chance of this happening is very small. This explains why simulations of CTMC models can run longer than those of PTA models.

The second difference is that the simulations of the PTA models start later than those of the CTMC. This is also due to the uniform distributions. For both simulations the parameters that were chosen gave each BE a lower clock bound of 7 days and an upper clock bound of 21 days. Meaning that the BE components could fail no later than 7 days. Because of the uniform distribution, the probability of failing before the lower bound and after the upper bound, is 0. This explains why in the simulations of the PTA start later. In fact, they start 7 simulated days later than the CTMC model. In this research, the effect of

changing the width of the uniform intervals is not investigated extensively and is a direction for future research. However, it has been observed that for small intervals, the simulations of the reliability become erratic as there is a large interval in which no transitions can take place. Furthermore, when the lower bound is reached, the probability of taking a transition in this interval is much higher, as opposed to a large interval.
The third difference is that the simulations of the PTA models gave a consistently higher chance of failure than the CTMC models. The maximum difference between the reliability of the PTA and the CTMC models for the first DFT is approximately 5%. For the second DFT this was approximately 2%. This could be due to the difference in the used probability functions. The CTMC models take the transition from the operational states to the failure states according to exponential distributions. The PTA models use uniform distributions for its transitions, whose bounds is determined by guards and invariants. It could be that the bounded uniform distribution causes the PTA models to take the transition from the operational states to the failure states earlier on average than the CTMC models, as it must take the transition before the upper clock bound has passed. The exponential probability distribution used by the CTMC models are not bounded and therefore can make the transition at any point in time. This might cause the CTMC models to take the transition to the non-operational state later in time than the PTA models on average. This would explain the lower reliability of the PTA models, but further research is required to confirm this.
A final observation that can be made, is that the reliability of the PTA models clearly shows the behaviour of an exponential distribution. However, the PTA models in this paper use uniform distributions to determine its failure and repair rates. Investigation of the mathematics behind this phenomenon is beyond the scope of this research, but could be interesting for future research.

5.3 Simulation Time

The results of the probability query and the simulation query for the basic DFT and those of the DFT with the spare-gates are shown in tables 1 and 3 and tables 2 and 4 respectively. The results show that the PTA models take more time to perform the queries than the CTMC model. The PTA models are approximately 0.006ms and 0.239s slower than the CTMC model for the probability and simulation query respectively. For the second DFT, the PTA models are respectively, 0.067ms and 0.675s slower. A reason for the PTA model being slower in verification for the DFT with the spares, could be the additional clocks, guards and invariants. It might be possible that these require more computation time than the exponential distributions of the CTMC models. PTA might be faster if a certain threshold of clocks, guards and invariants is met. When there are many clocks, the PTA model might be slower than the CTMC model.

6. CONCLUSION

The first research question that was asked in the introduction is: How does the simulated availability of each DFT differ for the PTA and CTMC models? According to the results of the simulations, no significant difference can be found between the PTA and CTMC models. Both formalisms yield approximately the same result of the availability of each DFT. The slight differences between both simulations are most likely from the use of probability functions, which cause two simulations of the same model to never be the same.
As to how the simulated reliability of each DFT differ for the PTA and CTMC models, there seems to be a difference. Simulations using PTA models generally lasted shorter than those of the CTMC models due to PTA models using bounded uniform distributions for the failure and repair rates. The second difference is that reliability of PTA models can only be simulated after the lowest lower bound of the uniform distributions has been reached in the simulation. Thirdly, the PTA models show a lower reliability than the equivalent CTMC models. This could be due to the difference in the used probability functions. Exponential distributions, used by the CTMC models, allow transitions to be taken at any point in time greater than 0. However, the uniform distributions, used by the PTA models, have a lower and upper bound determined by guards and invariants. The upper bound could cause PTA models to take the transition from the operational state to the failure state earlier in time on average than the CTMC models and thus yield a lower reliability. However, further research is required to confirm this.
The time each formalism takes to verify a system seems to depend on the formalism used. For the basic DFT and the DFT with the spare-gates, PTA models were slower. The reason that the PTA models were slower, could be due to the number of clocks, guards and invariants that are used. When there are many clocks to be considered in the PTA model, together with invariants and guards, PTA might be proven to be slower than CTMC. However, in this paper only two synthetic DFTs are considered. Analysing a real system and using PTA and CTMC with the additional dynamic gates in [2], might give more insight into the time PTA uses for verification of these systems in comparison to CTMC.
As mentioned in section 3, DFT extend fault trees by introducing dynamic gates such as the spare, PAND and FDEP gates. In this paper, however, only the spare-gate has been modelled with PTA and CTMC. This paper does not provide enough material to fully determine the usability of PTA as an approach to fault tree analysis. A possible direction for future research could be to investigate the usability of PTA regarding the other dynamic gates introduced by DFT. Furthermore, this paper analyses PTA for two synthetic DFTs. To fully determine the usability of PTA, they should also be analysed when used to model more complex, real systems. Other research directions include the investigation of the effects of changing the widths of the uniform intervals for the PTA models and an investigation of the reason behind the difference in reliability of the PTA and CTMC models. For now, it can be stated that a system whose upper and lower bounds for its failure and repair rates are known, can be modelled easily with the non-Markovian PTA with an uniform distribution, whereas CTMC can be used when only mean times of the failure and repair rates are available.

Acknowledgements

As the author of this paper, I would like to thank my supervisors for their invaluable feedback as well as the anonymous reviewers, who all contributed to the presentation of this paper. Furthermore, I would like to thank my family for their continued support.

7. REFERENCES

[1] P. Bell, S. Chen, and L. Jackson. On the decidability and complexity of problems for restricted hierarchical hybrid systems. *Theoretical Computer Science*, 652:47–63, 2016.

[2] H. Boudali, P. Crouzen, and M. Stoelinga. Dynamic fault tree analysis using input/output interactive markov chains. pages 708–717, 2007.

[3] F. Cassez, P. Jensen, and K. Guldstrand Larsen. Refinement of trace abstraction for real-time programs. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10506 LNCS:42–58, 2017.

[4] M. Čepin. *Binary Decision Diagram*, pages 101–112. Springer London, London, 2011.

[5] M. Jurdziński, F. Laroussinie, and J. Sproston. *Model Checking Probabilistic Timed Automata with One or Two Clocks*, pages 170–184. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[6] S. Krishna, L. Manasa, and A. Trivedi. What's decidable about recursive hybrid automata? pages 31–40, 2015.

[7] G. Norman, D. Parker, and J. Sproston. Model checking for probabilistic timed automata. *Formal Methods in System Design*, 43(2):164–190, 2013.

[8] G. Norman, D. Parker, and X. Zou. Verification and control of partially observable probabilistic systems. *Real-Time Systems*, 53(3):354–402, 2017. cited By 0.

[9] R. M. Sinnamon and J. Andrews. New approaches to evaluating fault trees. *Reliability Engineering & System Safety*, 58(2):89–96, 1997.

[10] J. Sproston. Probabilistic timed automata with clock-dependent probabilities. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10506 LNCS:144–159, 2017. cited By 0.

[11] B. Wu, M. Lemmon, and H. Lin. Formal methods for stability analysis of networked control systems with ieee 802.15.4 protocol. *IEEE Transactions on Control Systems Technology*, 2017. cited By 0; Article in Press.

[12] S. A. Zonouz and S. G. Miremadi. A fuzzy-monte carlo simulation approach for fault tree analysis. In *RAMS '06. Annual Reliability and Maintainability Symposium, 2006*, pages 428–433, Jan 2006.