2023 Design, Automation & Test in Europe Conference (DATE 2023)

Metric Temporal Logic with Resettable Skewed Clocks

Alberto Bombardelli
Fondaziome Bruno Kessler
Via Sommarive, 18, 38123 Povo TN
abombardelli@fbk.eu

Stefano Tonetta
Fondazione Bruno Kessler
Via Sommarive, 18, 38123 Povo TN
tonettas@fbk.eu

Distributed Real Time Systems (DRTS) are systems composed of various components communicating through a network and depending on a large number of timing constraints on the exchanged data and messages. Formal verification of DRTS is very challenging due to the intertwining of timing constraints and synchronization and communication mechanisms. Moreover, in a decentralized system, clocks may be skewed and it is necessary to synchronize them periodically, e.g., with the Berkeley synchronization algorithm.

In formal verification, local and global properties are typically specified in temporal logics such as Linear-time Temporal Logic (LTL) [9], which is able to specify temporal constraints on the succession of events or exchange of messages. When dealing with real-time systems and their properties, one of the most popular temporal logics is Metric Temporal Logics (MTL) [5], which enriches the temporal operators with bounds to constrain the time intervals in which formulas must be satisfied. Another variant, Event Clock Temporal Logic (ECTL) [10] uses event clock constraints to specify bounds on the time since the last time or until the next time a formula holds. Timed Propositional Temporal Logic (TPTL) [1], instead, uses freezing quantifiers to compare and constrain time at different points. One of the issues to specify and reason with MTL properties in DRTS is that clocks are not perfectly synchronized and the nodes of a distributed system may refer to different, possibly skewed, clocks. Distributed variants of MTL and ECTL use local temporal operators that refer to local times (e.g., [8]), which are independent but usually strictly increasing.

In this paper, we consider components of DRTS that use local clocks that are occasionally reset for synchronization and, so, that may be not monotonic. This may happen in practice, for example, when a component uses a local clock to send a message periodically and the clock is sometimes updated for synchronization with other components by means of a distributed algorithm for approximating real-value variables (cfr., e.g., [6]). Standard metric operators are not always suitable to express properties in this setting. For example, suppose to specify that a certain condition b holds for p time units with respect to a local clock c. If c is monotonic, this property can be formalized in a distributed version of MTL (as in [8] for ECTL) with the formula $G^c_{\leq p} b$. If c is not monotonic, the same formula requires b to hold in all (possibly disconnected) points that are less than p (see Figure 1), which is not what intended.

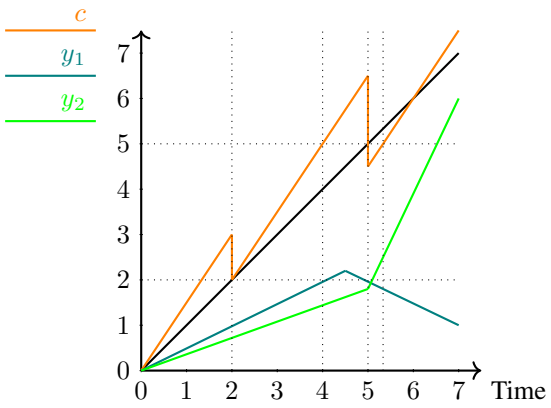

Fig. 1: Examples violating or satisfying $G^c_{\leq 5} b$ when the time reference is given by a skewed clock (orange line). The clock has a constant drift and is reset to approximately the correct value (black line). In order to satisfy $G^c_{\leq 5} b$, b must be true in the disconnected intervals $[0, 4]$ and $[5, 16/3]$. Thus, if b is $y_1 \leq 2$, then the formula $G^c_{\leq 5} b$ is satisfied, while if b is $y_2 \leq 2$, then the formula $G^c_{\leq 5} b$ is violated.

Thus, in this paper, we define alternative metric operators $\overline{U}$ and its derivates $(\overline{G}, \overline{F})$ with a semantics that is more suitable to specify properties of components that use skewed resettable clocks. For example, $\overline{G}^c_{\leq p} b$ requires a component to keep b true for the first p time units according to its local clock c, without relying on a clock reset, since this is not under its control.

MTL with skewed clocks and reset:: We first formally define some assumptions on the clocks to ensure that, despite the resets, they are diverging. We assume to be given two constants ϵ and λ that are used as bounds for the drift and resets, respectively.

Definition 1. *A "resettable skewed clock" (henceforward, simply, "clock") is a variable $c \in \mathcal{V}$ such that, for every trace π*

1) *for every $i \in \mathbb{N}$, $\pi(t)(c)$ is differentiable in I_i with $\frac{d\pi(t)(c)}{dt} \in [1-\epsilon, 1+\epsilon]$.*
2) *for every t, if t is a discrete step $|\pi(succ(t))(c)) - \nu(t)| \leq \lambda$.*

where $\nu(t)$ represents the value of time at step t and a discrete step is defined as a step in which time does not pass.

978-3-9819263-7-8/DATE23/© 2023 EDAA

Definition 2. *Given a signature Σ, a set of variables $\mathcal{V}$, and a set of clock variables $C \subseteq \mathcal{V}$, MTLSK formulas are built with the following grammar:*

$$\phi := pred \mid \phi \wedge \phi \mid \neg\phi \mid \phi \overline{U}^c_{\mathcal{I}} \phi \mid \phi U^c_{\mathcal{I}} \phi$$

where pred is a predicate over $\mathcal{V}$, $c \in C$, and $\mathcal{I} \subseteq \mathbb{R}$.

Abbreviations are defined similarly to the standard case.

The semantics of the new operator is defined as follows:

$$\begin{aligned} \pi, t \models \varphi \overline{U}^c_{\mathcal{I}} \psi \Leftrightarrow \text{exists } t' > t, \text{ s. t.} \\ \pi(t')(c) - \pi(t)(c) \in \mathcal{I}, \pi, t' \models \psi, \text{ and} \\ \text{for all } t \leq t'' < t: \\ \pi, t'' \models \varphi \text{ and } \pi(t'')(c) - \pi(t)(c) \in \mathcal{I}^- \end{aligned}$$

where $\mathcal{I}^- := \mathcal{I} \cup (-\infty, inf(\mathcal{I})]$.

$\overline{U}$ extends U to guarantee that in each point t'' between t and t', the difference between c at step t'' and c at step t is below the upper threshold of $\mathcal{I}$. Thus, surpassing the upper threshold of $\mathcal{I}$ without an occurrence of ψ falsifies $\varphi \overline{U}^c_{\mathcal{I}} \psi$ while it does not falsify $\varphi U^c_{\mathcal{I}} \psi$ since ψ might hold in a future point after a reset.

It should be noted that with resettable clocks, the interval domain is $\mathbb{R}$ instead of the positive counterpart used in MTL. It happens because clock resets may decrease clocks. Indeed, $\mathcal{I}$ is defined as a subset of $\mathbb{R}$ instead of $\mathbb{R}^+$.

Theorem 1. *For all π, φ: $\pi \models \varphi \overline{U}^c_{\mathcal{I}} \psi \Rightarrow \pi \models \varphi U^c_{\mathcal{I}} \psi$ and If $\sup(\mathcal{I}) = +\infty$, $\pi \models \varphi U^c_{\mathcal{I}} \psi \Leftrightarrow \pi \models \varphi \overline{U}^c_{\mathcal{I}} \psi$.*
If there is no reset (weakly monotonic case)
$\pi \models \varphi \overline{U}^c_{\mathcal{I}} \psi \Leftrightarrow \pi \models \varphi U^c_{\mathcal{I}} \psi$
Moreover, if there is no drift and no reset, i.e., $\epsilon = 0 \wedge \lambda = 0$ (perfect clocks), then

$$\pi \models \varphi \overline{U}^c_{\mathcal{I}} \psi \Leftrightarrow \pi \models \varphi U_{\mathcal{I}} \psi \Leftrightarrow \pi \models \varphi U^c_{\mathcal{I}} \psi$$

All proofs can be found in [2].

Compositional Reasoning Example: Consider for example a system of two components. The first component sends an alive signal (variable *alv*) to the second component unless there is a fault (variable f). The second component monitors the alive signal and, if absent, raises an alarm (variable *alm*). Globally, we expect that if there is a fault, an alarm is triggered in due time. The components use clocks c_1 and c_2, while the global clock is c. The compositional reasoning is formalized with the following formula: $(G(f \rightarrow G^{cl1}_{\leq p} \neg alv) \wedge G(G^{cl2}_{\leq p} \neg alv \rightarrow (F^{cl2}_{\leq p} alm))) \rightarrow G(f \rightarrow F^{cl}_{\leq p} alm)$

The formula is valid if the clocks are not skewed. Let us suppose instead that they are skewed and that the maximum drift between the local clocks and the global one is *r*. Then, the formula is valid if we add safe margins to the bounds to take into account the drift as follows: $(G(|c-c_2| \leq r) \wedge G(|c-c_1| \leq r) \wedge G(f \rightarrow \overline{G}^{cl1}_{\leq p} \neg alv) \wedge G(\overline{G}^{c_2}_{\leq p-4r} \neg alv \rightarrow (\overline{F}^{c_2}_{\leq p} alm))) \rightarrow G(f \rightarrow \overline{F}^{c}_{\leq p+2r} alm)$

Related work: Various works customized the modal operators of temporal logics to better suit the specification of DRTS. TPTL was extended in [12] by using explicitly multiple local clocks and supporting inequalities to express constraints on the precedence between local clock readings. In [8], a distributed variant of ECTL is proposed. Similarly, [7] defines a distributed modal logic where the time varies independently in each component of the system, represented by a network of timed automata. In all these works, local times are assumed strictly increasing, thus, not addressing the semantic issues of the temporal operators when the time is not monotonic.

The problem of modelling DRTS with drifting and synchronized clocks was considered in [11], where specific patterns of timed automata were proposed and verified. This work focuses on the modelling of clock drifts and synchronizations, but does not consider the specification of timed properties that refer to skewed synchronized clocks.

The satisfiability of MTL and TPTL over non-monotonic time has been studied in [3] in the context of data words, where timed words are considered a special case. However, the authors used the standard operators of MTL, without considering the semantic issues that we highlighted.

Last, we mention [4], which focuses on runtime verification of MTL formulas in a distributed system. Here, the authors address the problem of monitoring a global property on all traces that are compatible with a given sequence of local observations with timestamps taking into account the possible drift of local clocks. Thus, the metric operators are not, as in our case, used in local properties and related to local clocks.

REFERENCES

[1] R. Alur and T. A. Henzinger. A Really Temporal Logic. *J. ACM*, 41(1):181–204, 1994.

[2] A. Bombardelli and S. Tonetta. Metric Temporal Logic with Resettable Skewed Clocks - Extended version with proofs. Available at https://drive.proton.me/urls/D5Y7DARN3M#6io5si7nuH1E.

[3] C. Carapelle, S. Feng, O. F. Gil, and K. Quaas. Satisfiability for MTL and TPTL over Non-monotonic Data Words. In *LATA*, volume 8370 of *LNCS*, pages 248–259, 2014.

[4] R. Ganguly, Y. Xue, A. Jonckheere, P. Ljungy, B. Schornsteiny, B. Bonakdarpour, and M. Herlihy. Distributed Runtime Verification of Metric Temporal Properties for Cross-Chain Protocols. *CoRR*, abs/2204.09796, 2022.

[5] R. Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Syst.*, 2(4):255–299, oct 1990.

[6] N. A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.

[7] J. J. Ortiz, M. Amrani, and P. Schobbens. ML$_\mathcal{V}$: A Distributed Real-Time Modal Logic. In *NFM*, pages 19–35, 2019.

[8] J. J. Ortiz, A. Legay, and P. Schobbens. Distributed Event Clock Automata - Extended Abstract. In *CIAA*, pages 250–263, 2011.

[9] A. Pnueli. The temporal logic of programs. pages 46–57, 09 1977.

[10] J. Raskin and P. Schobbens. The Logic of Event Clocks - Decidability, Complexity and Expressiveness. *Journal of Automata, Languages and Combinatorics*, 4(3):247–286, 1999.

[11] G. Rodríguez-Navas and J. Proenza. Using Timed Automata for Modeling Distributed Systems with Clocks: Challenges and Solutions. *IEEE Trans. Software Eng.*, 39(6):857–868, 2013.

[12] F. Wang, A. K. Mok, and E. A. Emerson. Distributed Real-Time System Specification and Verification in APTL. *TOSEM*, 2(4):346–378, 1993.