Investigation on Time Properties of Timed-pNets

Yanwen Chen, Yixiang Chen, Eric Madelaine

Investigation on Time Properties of Timed-pNets

Yanwen CHEN [1,2,3], Yixiang CHEN [1], Eric MADELAINE [2,3] *

1 MoE Engineering Research Center for Software/Hardware Co-design Technology and Application,
East China Normal University, 200062, Shanghai, China
2 INRIA Sophia Antipolis Méditérannée, BP 93, 06902 Sophia Antipolis, France,
3 University of Nice Sophia Antipolis, CNRS, UMR 7271, 06900 Sophia Antipolis, France

Abstract. Timed-pNets is a semantic model to specify the communication behaviours of distributed systems. It has a tree style hierarchical structure. The leaves are timed specifications which consist of a set of logical clocks and clock relations. These logical clocks are encoded with delay variables and delay bound. In this paper we discuss how to detect time constraint conflicts and how to compute delay variables of clocks in the non-leaf nodes. Then we can check system's time properties like deadline. From our formalization of timed-pNets, we generate a system of logical clocks that can be simulated in the TimeSquare tool. We take a simple use case from ITS to simulate and check some time properties.

1 Introduction

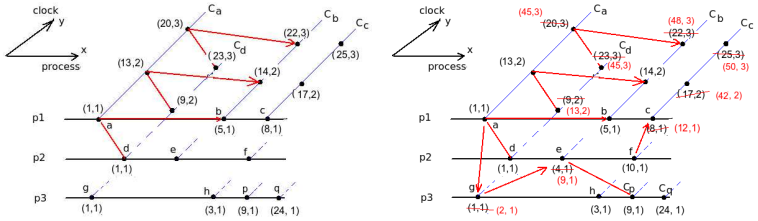
Timed-pNets[3] has been proposed to specify communication behaviours of heterogeneous distributed systems. This timed model is able to specify time constrains based on logical time. Logical time has proved its benefits in several domains. It was first introduced by Lamport to represent the execution of distributed systems [5]. Logical time can be multiform, a global partial order built from local total orders of clocks. The multiform nature of logical time consists in the ability to use any repetitive event as a reference for the other ones[2]. Inspired by the CCSL [1], we design clock relations to express the systems logical time constraints. So our model is a logical constraint model that is expressed by a set of logical clocks and clock constraints.

However, the logical constraint model cannot be directly used to check some time properties like deadline property since the distance of two clocks cannot be measured. In this paper, we focus on address the issue of transferring our logical constraints model to schedulable model in which the time properties like deadline can be checked. We propose a virtual timestamp to link those logical clocks. In Lamport, virtual time is identifiable by the succession of events (and therefore is

* This work was funded by DAESD of INRIA and ECNU; by NSFC (No. 61321064 and No. 61370100); by Shanghai Knowledge Service Platform Project (No. ZF1213).

discrete). It does not flow by its own means like real time whose passage we can not escape or influence. Based on his work, we define our virtual timestamp as two dimension values, and the number we assign to clocks as virtual timestamp is based on a reference clock. By this way, we can compare the delay variables from different logical clocks. Furthermore, we propose a theorems about computing delay variable of global clocks in timed-pNets. In the end, we take a use case from ITS (Intelligent Transportation Systems) for simulation.

2 Virtual Timestamps and Delay Bounds



(a) time diagram (b) updated time diagram

Fig. 1: Time Diagram and Updated One

We define a virtual timestamp as a pair of natural numbers: one represents when a timed-action occurs in terms of a reference clock (X-axis), another represents the order of the occurrences of a timed-action (Y-axis). Fig.1 shows us an example in which the timed-actions are assigned with virtual timestamps. In the figure, the processes are presented as solid black lines. The sequence of timed-actions executed in these processes are presented as solid black points on these black lines. The actions in each process are totally ordered. The communications between processes are represented by clock relations. For example, in the Fig. 1, the clock C_a and clock C_d are coincident. We use a sequence of red lines to represent the coincidence relation of two clocks. Similarly, we use a sequence of red arrows to represent the precedence relations (e.g. $C_a \prec C_b$). We define the virtual timestamps and their assignment rules as follows.

Definition 1 (Virtual Timestamps). *A virtual timestamp (denoted as $T(\alpha\_i)$) of a timed-action occurrence α_i is a pair of natural numbers (x_{α_i}, i) $(x_{\alpha\_i} \in \mathbb{N}, i \in \mathbb{N})$.*

Definition 2 (Virtual Timestamp Assignment Rules). *Let $T(\alpha\_i) \triangleq (x_{\alpha\_i}, i)$ be the virtual timestamp of the occurrence α_i of the clock C_α $(\alpha \in \mathcal{L}_{\mathcal{A},\mathcal{T},\mathcal{P}})$, and $T(\beta\_i) \triangleq (x_{\beta\_i}, i)$ be the virtual timestamp of the occurrence β_i of the clock C_β $(\beta \in \mathcal{L}_{\mathcal{A},\mathcal{T},\mathcal{P}})$. Then we have:*

- $C_\alpha = C_\beta \Rightarrow \forall i, x_{\alpha\_i} = x_{\beta\_i} := max(x_{\beta\_i}, x_{\alpha\_i})$
- $C_\alpha \prec C_\beta \Rightarrow \forall i, x_{\alpha\_i} < x_{\beta\_i}$ *and* $x_{\beta\_i} := max(x_{\alpha\_i}, x_{\beta\_i}) + t_{\beta_i}$ *(the variable t_{β_i} presents the delay time from the occurrence α_i to β_i in terms of the reference clock that a user chose. $t_{\beta_i} \geq 1, t_{\beta_i} \in \mathbb{N}$)*

Initially, for an independent clock (without any relation with other clocks), the X-axis value of the timestamps of the clock can be set with any natural number. The values will be updated according to the clock relations applied on this clock. Take the clock C_b in the Fig. 1(a) as an example, we know the relation $C_a \prec C_b \prec C_c$. According to the assignment rules, we must have $x_{a\_1} < x_{b\_1} < x_{c\_1}$. In this figure, the timestampe of the first occurrence of clock C_b is initially set as (5, 1). It can also be initially set as $(7, 1)$ only if it satisfied the condition $x_{a\_1} < x_{b\_1} < x_{c\_1}$. When adding new clock constraints, these timestamps in the Fig.1 may also be updated according to the assignment rules. For example, assume that we add other four clock relations ($C_a \prec C_g, C_g \prec C_e, C_f \prec C_c, C_e = C_p$). If the delay from C_f to C_c is 2, then the virtual timestamps may be updated by following the rules (as shown in the Fig.1(b)).

2.1 Time Constraint Conflicts

Since the timestamps may be updated, the clock delays may also be updated, which may cause time constraint conflicts. For example, in the Fig. 1, assume the delay bound of C_c is $[2, 5]$. Before we add the relation $C_f \prec C_c$, there is no time constraint conflict since $t_{C_c[1]} = 8 - 5 = 3 \in [2, 5]$. However, after adding this relation, we found out that $t_{C_c[1]} = 12 - 5 = 7 \notin [2, 5]$. Here we give a formal definition of time constraint conflicts.

Definition 3 (Time Constraints conflicts). *Let C_α be a clock built on timed-action $\alpha(p)^{t_\alpha|b_{t_\alpha}}$. A time constraint conflict of clock C_α exists if $\exists i \in \mathbb{N}$, $t_{\alpha_i} \notin b_{t_{\alpha_i}}$.*

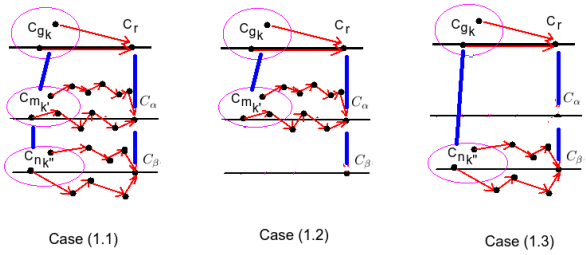


Fig. 2: Three cases in Theorem 1

2.2 Compute Delay and Delay Bounds

In timed-pNets, non-leaf nodes are the synchronization devices of their subsystems. The delays and delay bounds of the global logical clocks in these non-leaf nodes are computed in terms of the local logical clocks of the subsystems. When building these non-leaf nodes, time constraint conflicts may happen. Here, we propose a theorem to compute the delays and delay bounds of these global clocks so that we can check if time constraint conflicts exist.

Theorem 1 (The Delay Bounds of Global Clocks). *Given a timed-pNet $< P, A_G, C_G, J, \widetilde{A}_J, \widetilde{C}_J, \widetilde{R}_J, \overrightarrow{V} >$. Assume that all local clocks (in the set $\widetilde{C}_J$)*

have no time constraint conflict. Consider a global clock C_γ *and let* $\mathbb{C}_g = \{C_{g_k}\}(k \in \mathbb{N})$ *be the set of causal clocks of* C_γ *(*$\mathbb{C}_g \subseteq C_G$, $C_\gamma \in C_G$, $\gamma \triangleq \gamma(p_\gamma)^{t_\gamma|b_{t_\gamma}}$*).*

(1) When $\overrightarrow{v} =< ..., C_\alpha, ..., C_\beta, ... >\rightarrow C_\gamma$*. As shown in Fig. 2, let* $\mathbb{C}_m = \{C_{m_{k'}}\}(k' \in \mathbb{N})$ *be a set of local clocks that are in the same hole as* C_α*, and that contribute to generate the global clocks in* $\mathbb{C}_g$*. Let* $\mathbb{C}_n = \{C_{n_{k''}}\}$ *be a set of local clocks that are in the same hole as* C_β*, and that contribute to generate the global clocks also in* $\mathbb{C}_g$*.*

(1.1) If $< \ldots, C_{m_{k'}}, \ldots, C_{n_{k''}}, \ldots >\rightarrow C_{g_k}$ *as shown the case (1.1) in Fig. 2, then*
$b_{C_\gamma} = [min\{min\{l(b_{P_{C_{m_{k'}}\rightarrow C_\alpha}})|k' \in \mathbb{N}\}, min\{l(b_{P_{C_{n_{k''}}\rightarrow C_\beta}})|k'' \in \mathbb{N}\}\}, max\{max\{u(b_{P_{C_{m_{k'}}\rightarrow C_\alpha}})|k' \in \mathbb{N}\}, max\{u(b_{P_{C_{n_{k''}}\rightarrow C_\beta}})|k'' \in \mathbb{N}\}\}]$ *(*$C_{m_{k'}} \in \mathbb{C}_m, C_{n_{k''}} \in \mathbb{C}_n, k', k'' \in \mathbb{N}$*);*

(1.2) If $< \ldots, C_{m_{k'}}, \ldots, \ldots, \ldots >\rightarrow C_{g_k}$ *as shown the case (1.2) in Fig. 2, then*
$b_{C_\gamma} = [min\{l(b_{P_{C_{m_{k'}}\rightarrow C_\alpha}})|k' \in \mathbb{N}\}, max\{u(b_{P_{C_{m_{k'}}\rightarrow C_\alpha}})|k' \in \mathbb{N}\}]$ *(*$C_{m_{k'}} \in \mathbb{C}_m, k' \in \mathbb{N}$*),*

(1.3) If $< \ldots, \ldots, \ldots, C_{n_{k''}}, \ldots >\rightarrow C_{g_k}$ *as shown the case (1.3) in Fig. 2, then*
$b_{C_\gamma} = [min\{l(b_{P_{C_{n_{k''}}\rightarrow C_\beta}})|k'' \in \mathbb{N}\}, max\{u(b_{P_{C_{n_{k''}}\rightarrow C_\beta}})|k'' \in \mathbb{N}\}]$ *(*$C_{n_{k''}} \in \mathbb{C}_n, k'' \in \mathbb{N}$*),*

(2) When $\overrightarrow{v} =< ..., C_\alpha, ..., \ldots, ... >\rightarrow C_\gamma$*. Let* $\mathbb{C}_m$ *be a set of local clocks that in the same hole as* C_α*, and that contribute to generate the global clocks in* $\mathbb{C}_g$*. Then* $b_{C_\gamma} = [min\{l(b_{P_{C_{m_{k'}}\rightarrow C_\alpha}})|k' \in \mathbb{N}\}, max\{u(b_{P_{C_{m_{k'}}\rightarrow C_\alpha}})|k' \in \mathbb{N}\}]$ *(*$C_{m_{k'}} \in \mathbb{C}_m, k' \in \mathbb{N}$*).*

Because of the page limitation, we omit the proof. In the theorem, $l(b_{P_{C_{m_{k'}}\rightarrow C_\alpha}})$ (resp.$u(b_{P_{C_{m_{k'}}\rightarrow C_\alpha}})$) means the lower (resp. upper) bound of $b_{P_{C_{m_{k'}}\rightarrow C_\alpha}}$ from clock $C_{m_{k'}}$ to clock C_α along a set of paths.

3 Simulation

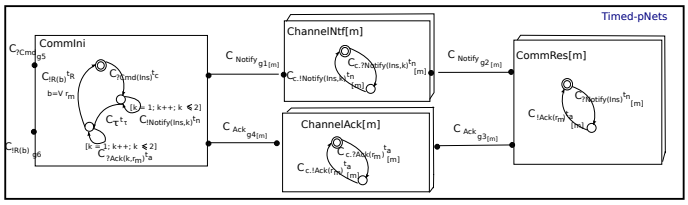


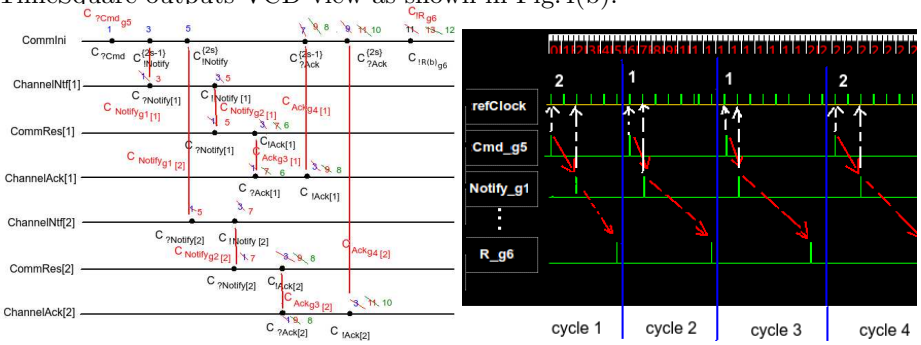
Fig. 3: Example of a Timed-pNets subsystem

We simulate the system of the Fig. 3 by means of the TimeSquare tool [4] to check system time constraint conflicts. For simplification, we choose a reference clock that ticks periodically. All delays and delay bounds of other logical clocks are specified in terms of this reference clock. In our simulation, we assume that

the delay bounds of all action occurrences are between $[1, 3]$ in the sense that the delays of those actions should stay between the first and the third occurrences of the reference clock. The simulation result tells us if conflicts exist in the system.

Result: TimeSquare reports us an error. The main reason is that the communications between those component create conflicts. By analyzing those updated virtual timestamps in the Fig.4(a), we can see that a time constraint conflict happens on the clock $C_{?Ack}^{\{2s-1\}}$ $(x_{C_{?ack}^{\{2s-1\}}[1]} - x_{C_{!notify}^{\{2s\}}[1]} = 9 - 5 = 4 \notin [1, 3])$.

Solution: To fix the issue, we set the delay of $?Notify\_i$ in component "CommRes" to 1 and limit the delays of all clocks less than 2 except the clock $C_{?Ack}^{\{2s-1\}}$. After redoing the simulation, we found out that no conflict exists. TimeSquare outputs VCD view as shown in Fig.4(b).



(a) Time Constraint Conflicts (b) Checking property

Fig. 4: Conflict Analysis and Results

4 Conclusion

In this paper, we investigated the time constraints and properties of timed-pNets model. We took a use case from ITS to build a timed-pNets communication model. From our formalization of timed-pNets, we generated a system of logical clocks and solved the issue of detecting time constraint conflicts.

References

1. Charles André. Syntax and Semantics of the Clock Constraint Specification Language (CCSL). Rapport de recherche RR-6925, INRIA, 2009.
2. Frédéric Boussinot and Robert De Simone. The esterel language. *Proceedings of the IEEE*, 79(9):1293–1304, 1991.
3. Yanwen Chen, Yixiang Chen, and Eric Madelaine. Timed-pNets: A formal communication behavior model for distributed systems. In *Journal of Frointier of Computer Science*, 2014.
4. Julien Deantoni and Frédéric Mallet. TimeSquare: Treat your Models with Logical Time. volume 7304 of *Lecture Notes in Computer Science - LNCS*, pages 34–41, Prague, Tchèque, République, May 2012. Springer.
5. Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, 1978.