Parameterized Verification of Timed Networks with Lossy Broadcast and Clock Invariants

Étienne André[1], Paul Eichler[2], Swen Jacobs[2],
Shyam Lal Karra[2], and Ocan Sankur[3]

[1] Université Sorbonne Paris Nord, LIPN, CNRS UMR 7030, France
[2] CISPA Helmholtz Center for Information Security, Germany
[3] Université Rennes, CNRS, France

Keywords: Networks of Timed Automata, Parameterized Verification

1 Extended Abstract

Formally reasoning about concurrent systems is difficult, in particular if correctness guarantees should hold regardless of the number of interacting processes—a problem that is also known as *parameterized verification* [2, 7], since the number of processes is considered a parameter of the system. Parameterized verification is undecidable in general [12] and even in very restricted settings, e.g., for safety properties of systems composed of finite-state processes with rather weak communication primitives, such as token-passing or transition guards [24, 18]. However, a long line of research has striven to identify classes of systems and properties for which parameterized verification problems are decidable [18, 22, 19, 20, 17, 16, 9, 21], usually with finite-state processes.

Timed automata (TAs) [8] provide a computational model that combines real-time constraints with concurrency, and are therefore an expressive and widely used formalism to model real-time systems. However, TAs are usually used to model a constant and *fixed* number of system components. When the number of components or agents (e.g., nodes in a network, voters in an election protocol, etc.) is very big or unknown, considering the static combination of n agents becomes highly impractical, or even impossible if n is unbounded. There is a line of research into networks of arbitrary numbers of timed components (see e.g., [6, 15, 4, 10]). However, due to the expressiveness of TAs, the results in this area are often negative, or limited to severely restricted cases such as TAs without clock invariants (that could force a process to leave a location) and with a single clock [6, 5, 4, 1, 3].

A system model that has received some attention recently is that of *Disjunctive Timed Networks* (DTNs), which combines the very expressive formalism of TAs with the relatively weak communication primitive of *disjunctive guards* [18]: transitions can be guarded with a location (called "guard location"), and such a transition can only be taken by a TA in the network if (at least) one other process is in that location upon firing. For example, in the DTN in Fig. 1, the

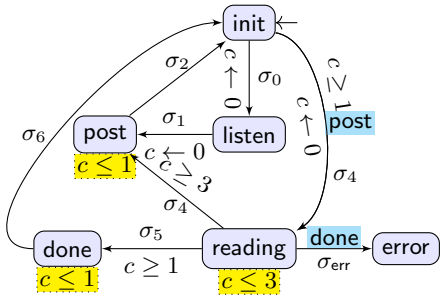

Fig. 1: Asynchronous data read example

transition from init to reading is guarded by location post: if a process in location init wants to reach reading, then at least one other process must be in post at the same time. A related communication primitive is *lossy broadcast* [17], which consists in one process taking a sending transition, and an arbitrary subset of the potential receivers synchronizing with a corresponding transition. For finite-state processes, it is known that lossy broadcast is strictly more expressive than disjunctive guards [14], but for timed processes the relation was unknown until now.

Parameterized model checking of DTNs has first been studied in [23], who considered local trace properties in the temporal logic MTL, and showed that the problem can be solved with a statically computable cutoff, i.e., a number of processes that is sufficient to determine satisfaction in networks of any size. However, their result is restricted to the case when guard locations do not have location invariants, and they showed that statically computable cutoffs do *not exist* for the case when TAs can have location invariants on all locations. However, the non-existence of such cutoffs does not in general imply that the problem is undecidable.

In the first part of this work [11], we show how to circumvent the expensive construction of a cutoff system for the case without invariants, instead using a modified zone graph algorithm. This allows us to construct a *summary automaton* that has the same language as a single TA in a network with arbitrarily many processes, and thus enables parameterized model checking of local safety and liveness properties. Moreover, we identify sufficient conditions on the TAs which imply correctness of this approach even in the presence of invariants. However, these conditions are semantic, and it is not obvious how to build models that satisfy them; e.g., the TA in Fig. 1 does not satisfy them.

In the second part, we show that, surprisingly, and despite the absence of cutoffs [23], the parameterized model checking problem for local safety properties is decidable for DTN in the general case, without any restriction on location invariants; we give an EXPSPACE algorithm for reachability properties. The technique circumvents the non-existence of cutoffs by constructing a modified region automaton that directly takes communication via disjunctive guards into

account. However, we note that this construction is in general not correct for liveness properties.

We also extend our setting to the more difficult problem of global reachability properties, which (including deadlock detection as a special case) is often considered in parameterized verification of untimed systems [22, 18, 17, 13], but to the best of our knowledge has not been considered in any of the works on networks of TAs. We show how our algorithm can be extended to such properties, at the cost of an exponential blowup.

Finally, we establish a result that has an independent interest. We prove that communication by lossy broadcast is equivalent to communication by disjunctive guards for timed automata *with* location invariants (while the equivalence does not hold without clocks [14]). Thus our algorithms also cover lossy broadcast with invariants, which might also explain their higher complexity with respect to the case of disjunctive guards without invariants [11].

We have thus identified a powerful yet decidable formalism for distributed real-time systems communicating by lossy broadcast or by location guards.

References

1. Abdulla, P.A., Atig, M.F., Cederberg, J.: Timed lossy channel systems. In: D'Souza, D., Kavitha, T., Radhakrishnan, J. (eds.) FSTTCS. LIPIcs, vol. 18, pp. 374–386. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2012). https://doi.org/10.4230/LIPICS.FSTTCS.2012.374
2. Abdulla, P.A., Delzanno, G.: Parameterized verification. International Journal on Software Tools for Technology Transfer **18**(5), 469–473 (2016). https://doi.org/10.1007/s10009-016-0424-3
3. Abdulla, P.A., Delzanno, G., Rezine, O., Sangnier, A., Traverso, R.: On the verification of timed ad hoc networks. In: Fahrenberg, U., Tripakis, S. (eds.) FORMATS. Lecture Notes in Computer Science, vol. 6919, pp. 256–270. Springer (2011). https://doi.org/10.1007/978-3-642-24310-3˙18
4. Abdulla, P.A., Delzanno, G., Rezine, O., Sangnier, A., Traverso, R.: Parameterized verification of time-sensitive models of ad hoc network protocols. Theoretical Computer Science **612**, 1–22 (2016). https://doi.org/10.1016/j.tcs.2015.07.048
5. Abdulla, P.A., Deneux, J., Mahata, P.: Multi-clock timed networks. In: LiCS. pp. 345–354. IEEE Computer Society (2004). https://doi.org/10.1109/LICS.2004.1319629
6. Abdulla, P.A., Jonsson, B.: Model checking of systems with many identical timed processes. Theoretical Computer Science **290**(1), 241–264 (2003). https://doi.org/10.1016/S0304-3975(01)00330-9
7. Abdulla, P.A., Sistla, A.P., Talupur, M.: Model checking parameterized systems. In: Clarke, E.M., Henzinger, T.A., Veith, H., Bloem, R. (eds.) Handbook of Model Checking, pp. 685–725. Springer (2018). https://doi.org/10.1007/978-3-319-10575-8˙21
8. Alur, R., Dill, D.L.: A theory of timed automata. Theoretical Computer Science **126**(2), 183–235 (Apr 1994). https://doi.org/10.1016/0304-3975(94)90010-8
9. Aminof, B., Kotek, T., Rubin, S., Spegni, F., Veith, H.: Parameterized model checking of rendezvous systems. Distributed Computing **31**(3), 187–222 (2018). https://doi.org/10.1007/s00446-017-0302-6

10. André, É., Delahaye, B., Fournier, P., Lime, D.: Parametric timed broadcast protocols. In: Enea, C., Piskac, R. (eds.) VMCAI. Lecture Notes in Computer Science, vol. 11388, pp. 491–512. Springer (2019). https://doi.org/10.1007/978-3-030-11245-5˙23
11. André, É., Eichler, P., Jacobs, S., Karra, S.L.: Parameterized verification of disjunctive timed networks. In: Dimitrova, R., Lahav, O. (eds.) VMCAI. Lecture Notes in Computer Science, vol. 14499, pp. 124–146. Springer (2024). https://doi.org/10.1007/978-3-031-50524-9˙6
12. Apt, K.R., Kozen, D.: Limits for automatic verification of finite-state concurrent systems. Information Processing Letters **22**(6), 307–309 (1986). https://doi.org/10.1016/0020-0190(86)90071-2
13. Balasubramanian, A.R., Esparza, J., Lazic, M.: Complexity of verification and synthesis of threshold automata. In: Hung, D.V., Sokolsky, O. (eds.) ATVA. Lecture Notes in Computer Science, vol. 12302, pp. 144–160. Springer (2020). https://doi.org/10.1007/978-3-030-59152-6˙8
14. Balasubramanian, A.R., Weil-Kennedy, C.: Reconfigurable broadcast networks and asynchronous shared-memory systems are equivalent. In: Ganty, P., Bresolin, D. (eds.) GandALF. EPTCS, vol. 346, pp. 18–34 (2021). https://doi.org/10.4204/EPTCS.346.2
15. Bertrand, N., Fournier, P.: Parameterized verification of many identical probabilistic timed processes. In: Seth, A., Vishnoi, N.K. (eds.) FSTTCS. LIPIcs, vol. 24, pp. 501–513. Schloss Dagstuhl – Leibniz-Zentrum für Informatik (2013). https://doi.org/10.4230/LIPIcs.FSTTCS.2013.501
16. Bloem, R., Jacobs, S., Khalimov, A., Konnov, I., Rubin, S., Veith, H., Widder, J.: Decidability of Parameterized Verification. Synthesis Lectures on Distributed Computing Theory, Morgan & Claypool Publishers (2015). https://doi.org/10.2200/S00658ED1V01Y201508DCT013
17. Delzanno, G., Sangnier, A., Zavattaro, G.: Parameterized verification of ad hoc networks. In: Gastin, P., Laroussinie, F. (eds.) CONCUR. Lecture Notes in Computer Science, vol. 6269, pp. 313–327. Springer (2010). https://doi.org/10.1007/978-3-642-15375-4˙22
18. Emerson, E.A., Kahlon, V.: Reducing model checking of the many to the few. In: McAllester, D.A. (ed.) CADE. Lecture Notes in Computer Science, vol. 1831, pp. 236–254. Springer (2000). https://doi.org/10.1007/10721959˙19
19. Emerson, E.A., Namjoshi, K.S.: On reasoning about rings. International Journal of Foundations of Computer Science **14**(4), 527–550 (2003). https://doi.org/10.1142/S0129054103001881
20. Esparza, J., Finkel, A., Mayr, R.: On the verification of broadcast protocols. In: LiCS. pp. 352–359. IEEE Computer Society (1999). https://doi.org/10.1109/LICS.1999.782630
21. Esparza, J., Jaax, S., Raskin, M.A., Weil-Kennedy, C.: The complexity of verifying population protocols. Distributed Computing **34**(2), 133–177 (2021). https://doi.org/10.1007/s00446-021-00390-x
22. German, S.M., Sistla, A.P.: Reasoning about systems with many processes. Journal of the ACM **39**(3), 675–735 (1992). https://doi.org/10.1145/146637.146681
23. Spalazzi, L., Spegni, F.: Parameterized model checking of networks of timed automata with Boolean guards. Theoretical Computer Science **813**, 248–269 (2020). https://doi.org/10.1016/j.tcs.2019.12.026
24. Suzuki, I.: Proving properties of a ring of finite-state machines. Information Processing Letters **28**(4), 213–214 (1988). https://doi.org/10.1016/0020-0190(88)90211-6