Synthesis of Extremely Large Time-Triggered Network Schedules †

Francisco Pozo

School of Innovation, Design and Engineering, Mälardalens University, 721 23 Västerås, Sweden; francisco.pozo@mdh.se; Tel.: +46-736-620-610

† Presented at the IS4SI 2017 Summit DIGITALISATION FOR A SUSTAINABLE SOCIETY, Gothenburg, Sweden, 12–16 June 2017

Abstract: Synthesis of time-triggered network schedules is a known practice to obtain low jitter and bounded end-to-end delay needed in some applications where not only the information but the transmission timings are crucial. However, synthesizing such schedules is a NP-complete problem where state-of-the-art synthesizers are being out scaled by the increasing size and complexity of new application networks. We present different divide and conquer approaches to handle the increasing complexity of future networks while maintaining the synthesis time reasonably low.

Keywords: real-time networks; time-triggered; scheduling; SMT solvers

1. Context

Time-Triggered switched Ethernet networks are increasing in size and complexity as they are planning to be deployed in large industrial networks such as mega-factories, and projected to be used in the future in smart cities. Traffic in time-triggered networks follows an offline schedule designed beforehand that contains the transmission times of all the time-triggered frames through the links in the network. This is important as any timing deviation could cause a major failure in the whole system. However, synthesizing this schedule means a major challenge when adapting time-triggered network in a larger scale. Schedule synthesis is a well-known NP-complete problem with complexity driven by the network size and the number of frames it contains. State-of-the-art schedulers can synthesize such schedules in a reasonable amount of time, but, they start to present scalability issues and are not able to cope with the size and complexity introduced by future applications.

In addition, applications are starting to require wireless capabilities in some areas of its networks, introducing a mixture of wired and wireless communication in the same network. Wireless communication also increases the complexity in schedule synthesis as the difference in transmission speeds between wired and wireless links is too significant to apply typical complexity reduction techniques, such as raster [1].

2. Research Goals

The goal of this research is to provide a new approach that can overcome the scalability issues of current schedule synthesis approaches before the deployment of the network. In general, a good strategy in engineering to cope with large problems is to use divide and conquer approach. In the case of schedule synthesis, we divide the schedule in smaller schedules, called segments, in which every segment will be a non-overlapping time interval of the complete schedule. Once all segments are defined, we allocate as many frames as we can using a state-of-the-art scheduler and concatenate them to obtain the final schedule. In our case, we apply Satisfiability Modulo Theories (SMT) solvers [2], that can find the satisfiability of a set of constraints and, if exists, provide an example of such satisfiability, in our case, frames allocated in a segment. However, some of the frames are dependent

between them, and present inter-segment constraint that go beyond single segments. These inter-segment constraints bring a challenge to our approach as each segment is scheduled independently and there is no mechanism to satisfy constraints outside the segment being scheduled. To overcome this limitation, we release the SMT solver of such constraints, and handle them in our algorithms with the selection of frames and the addition of some extra constraints in each segment for the inter-segment constraints to be satisfied.

3. Results Obtained

We evaluated our approach using the topology of a network deployed to provide free Internet to a large touristic area and creating synthetic time-triggered traffic. This network, that we call Actual, consist in 81 end systems that exchange information between them through 44 switches and 248 links. Then, we created two new larger networks, Large, that consists is 5 times bigger than the Actual network, and Very Large, which is 11 times bigger. We provided between 5000 and 50,000 synthetic frames to these networks and synthesize its schedules with our approach using the SMT solver Yices 2.4 [3] to allocate frames into the segments.

The synthesis times can be seen in Figure 1. For the Actual size network, we can schedule 5000 frames in one minute and up to 50,000 frames in less than one hour. This is a huge performance and scalability improvement, as state-of-the-art synthesizers could only schedule up to 1000 frames in 20 min for smaller networks that the used in our evaluation. The synthesis time for Large and Very Large networks increases due to its larger complexity, but the synthesizer is still able to find valid schedules. It is important to note that the increase in synthesis time for networks with more frames is almost linear instead of exponential. This allows us to, if it is needed, to schedule networks with even more frames without experiencing scalability issues, which was the main problem of previous approaches.
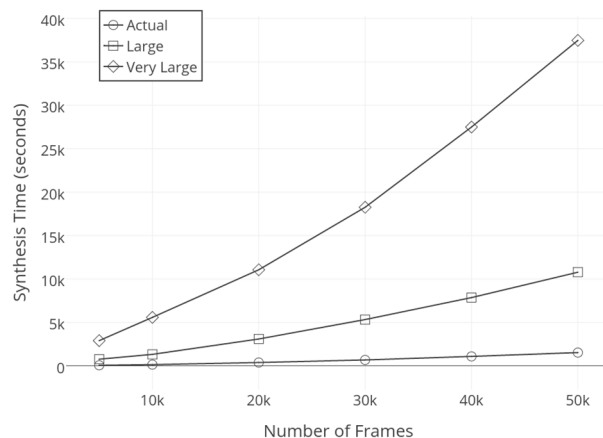

Figure 1. Synthesis time in seconds for different networks sizes and different number of frames.

4. Next Steps

Instead of keep pursuing an increase of performance in the schedule synthesis, we would like to abstract and automatize the process of constraint selection for the segments. In our research, we found that the selection of constraints, the number and how to add extra constraints in every segment has a great impact in the performance of the synthesizer. We would like to automatize this problem with the use of machine learning to find better segments and constraints division. This may allow us to use our approach to solve scheduling problem outside Time-Triggered networks if they are formulated as SMT constrained problems. It also has a good impact in our performance, as it could allow us to find segments that can be scheduled in parallel.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Mok, A.K.; Wang, W. Window Constrained Real-Time Periodic Task Scheduling. In Proceedings of the 22nd IEEE Real-Time Systems Symposium (RTSS 2001), London, UK, 3–6 December 2001; pp. 15–24.
2. Ranise, S.; Tinelli, C. Satisfiability Modulo Theories. *IEEE Intell. Syst. Mag. Trends Controv.* **2006**, *21*, 71–81.
3. Durertre, B. Yices 2.2. In *Computer Aided Verification*; Springer: Vienna, Austria, 2014; pp. 737–744.