Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

TOWARDS A NOVEL PROTOCOL FOR EFFICIENT CONTROL AND USAGE OF TRAFFIC SIMULATION

Zhuoxiao Meng

Department of Informatics
Technical University of Munich
Boltzmannstr. 21
Garching, 85748, GERMANY

Intelligent Cloud Technologies Laboratory
Huawei Munich Research Center
Riesstr. 25
Munich, 80992, GERMANY

ABSTRACT

In this work, we propose the *Traffic Simulation Control Protocol* (TraSCoP), a universal protocol for traffic simulation systems, allowing efficient simulation of traffic management systems by controlling running traffic simulations from external connected applications. Compared to existing approaches, TraSCoP enables the direct retrieval of temporally aggregated data with an event-driven control architecture, resulting in a substantial reduction in the communication overhead and hence a better simulation performance. A proof-of-concept study on the simulation of an adaptive traffic light control system demonstrates a 4.5 times speed-up using TraSCoP compared to TraCI, a widely used protocol for controlling traffic simulators.

1 INTRODUCTION AND MOTIVATION

Traffic management systems are increasingly trending towards complexity, dynamics, and adaptiveness. Existing traffic simulators are often not adequate for the simulation purpose. In contrast to constantly extending the functionality of a mature simulator, coupling it with other applications to compose an integrated simulation framework, can not only broaden the applicability of a traffic simulator, but also offer better reusability and modularity. In this context, traffic simulation requires runtime controllability, allowing for instance the collection of traffic statistics and the deployment of management policies during the execution.

At present, a widely used publicly available protocol for runtime controllability of traffic simulations is the *Traffic Control Interface* (TraCI) (Wegener et al. 2008). However, due to the lack of support for temporally aggregated traffic statistics (e.g., lane volumes and fuel consumption), TraCI is inefficient for simulating traffic management systems. Due to the synchronous execution mode, each TraCI request concerns only a single time step and the simulation must be suspended while requests are processed. Therefore, for temporally aggregated data, clients have to periodically communicate with the running simulation to read the current simulation state. This creates considerable amounts of communication overhead, substantially slowing down the simulation.

We are working on a novel protocol: TraSCoP, which enables the direct retrieval of temporally aggregated data at runtime with an event-driven approach. It is efficient and particularly suitable for simulations of traffic management systems. We also aim to make the protocol generic enough and easy to implement so that the maximum possible scenarios can be addressed with minimal implementation effort.

2 APPROACH

Similar to TraCI, with TraSCoP, the traffic simulator acts as a server, waiting for requests from external applications (clients) and responding accordingly. Each pair of request and response represents an operation of simulation control: it can either be data retrieving, data updating, or an execution-related action (e.g., pause simulation). In distinction to TraCI, TraSCoP requests are processed asynchronously with regard to the simulation. Temporal queries explicitly include the time (interval) of interest (rather having to issue a query at the time of interest). This enables directly querying temporally aggregated data, such as lane volumes from 8:00 to 9:00.

When the simulation receives such a request, it first wraps the request into a control-event and schedules it into the core logic of the simulation update. Meanwhile, the simulation continues running. For the given example, at 8:00, this event is executed, telling the data processor embedded inside the simulator to start gathering relevant data until simulation runs to 9:00. Afterwards, the collected data is processed as required and sent back as a reply. However, what happens if, the time value of this request has already elapsed when the simulation receives it? Since this request was received too late to be served, it is defined as invalid and the scheduling of its control-event will thus fail. A failure message will be sent back to the client and the session for this request will be closed. This intermediate message can help the client to better organize the rest of its control requests, e.g., an exception can be thrown when client receives a failure message.

With this event-driven approach, the simulation can run continuously in the time gap between the request receiving and the response sending. This sets the ground for collecting temporal aggregated data. Nevertheless, during the time gap, the traffic simulators must remain open to upcoming new messages, i.e., the simulation server should not be blocked due to pending requests. Therefore, an asynchronous communication framework is used to handle the information exchange.

3 PRELIMINARY RESULTS

To test TraSCoP with a real-world application, we simulated a simple adaptive traffic light control system, which assumes vehicle detectors are placed upstream of intersections to measure approach flows for each cycle. Therefore we implemented a client, which continuously queries the lane volumes for a traffic light cycle at each entrance lane of each intersection. A grid road network with 100 intersections, each with 12 entrance lanes and a fixed 200 seconds traffic light cycle, is taken as input. We implemented TraSCoP on top of CityMos (Zehe et al. 2017) using Google's Protocol Buffers and gRPC as communication framework.

As a baseline, simulating one day traffic with this scenario without any runtime interaction takes 324 seconds. With TraCI, we used subscriptions to query data for optimal performance: A client initially subscribes to each entrance lane and receives messages with data periodically (i.e. one message per lane per second). The client then computes the temporally aggregated statistics from the ”raw” data it received. This results in a total of 103 680 000 messages sent throughout the whole simulation. In the end, the simulation consumes 2802 seconds, nearly 10 times slower than the baseline due to communication overhead.

With TraSCoP, the communication overhead can be substantially reduced. Here, client sends one request for each cycle of each intersection, taking the cycle time as the time interval and the entrance lanes together in an array as the query objects. Only 43 200 queries are thus required for the data retrieval during the entire simulation. This further leads to a reduction of runtime to 630 seconds, i.e., only 2 times slower when compared to the baseline.

We are confident that TraSCoP can serve as a universal tool for the traffic simulation community, taking the use and control of traffic simulation to a new level. In future work, we will refine the details of the protocol and enrich the request types to cover the needs of most scenarios and real-world applications. TraSCoP will be made publicly available under a non-restrictive license.

REFERENCES

Wegener, A., M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux. 2008, April. “TraCI: an interface for coupling road traffic and network simulators”. In 11th Communications and Networking Simulation symposium (CNS 2008), 155–163: ACM.

Zehe, D., S. Nair, A. Knoll, and D. Eckhoff. 2017. “Towards CityMoS: A Coupled City-Scale Mobility Simulation Framework”. In 5th GI/ITG KuVS Fachgespräch Inter-Vehicle Communication (FG-IVC 2017): Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU).