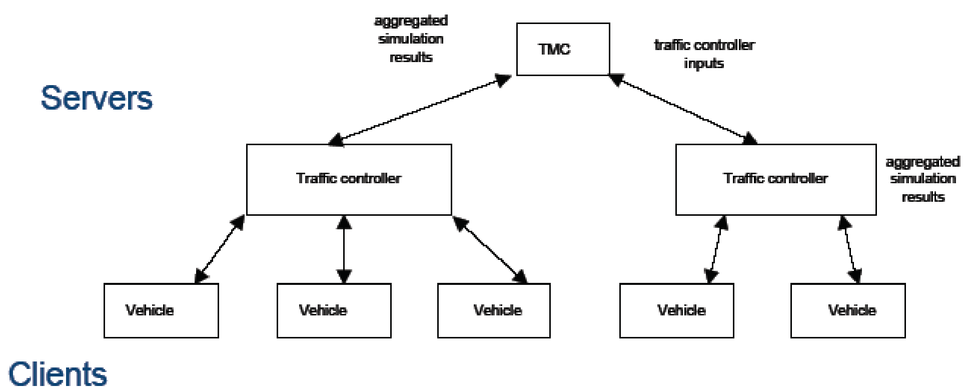
A report on "Dynamic Data Driven Application Simulation of Surface Transportation Systems - R. Fujimoto et al."

Soham Chakraborty

Summary

The paper describes how Dynamic Data Driven Application Simulation can be used to manage surface transportation. It proposes a multi layered architecture, consisting of a Regional Traffic Management Center (TMC) at the top of the hierarchy. The intermediate layer is composed of *computer systems housed in traffic signal controllers and other roadside equipment* [1]. Fixed sensors and vehicle mounted sensors and computer systems form the end nodes of the hierarchy.



The vehicles download maps of its region of interest, from the server, as it is traveling along a road. This is cached for future reference. It then gathers real time information about the traffic (such as flow and density) from its in-vehicle sensors, road based (fixed) sensors and other similar vehicles(if any) in vicinity. It can also gather aggregate data about *vehicle density and traffic flow rates from the server* [1]. All this data is used to run simulations in the on-board computer systems. Vehicle based simulations will typically be used for the purposes of *route planning and travel time prediction* [1]. The "decision maker" module of the computer system will use the results of the simulation will make decisions regarding the optimal route between two points. The information gathered by the on-board sensors and the simulation results are passed on to the Traffic Controller Computers.

The Traffic Controllers, obtain real time data about the traffic from both fixed sensors and vehicle based systems. It also collects the results of the simulations from the vehicle based computers. Traffic Controller s*imulations are used for purposes such as adjusting signal timings* [1]. Since the region under surveillance by the sensors overlap, the Traffic Controller also aggregates the results from these sensors and resolves conflicts (if any). The aggregate real time traffic data and the results from the various simulations are used by the TMC computers for city-wide traffic monitoring and management.

The above systems are termed *ad hoc distributed simulations because they are composed of loosely coupled, largely autonomous simulations, each modeling a different portion of the overall system, often at different resolutions* [1]. Ad hoc distributed simulations often model regions that overlap and hence require conflict resolving mechanism, in cases where two or more autonomous systems (observing the same region) arrive at different results.

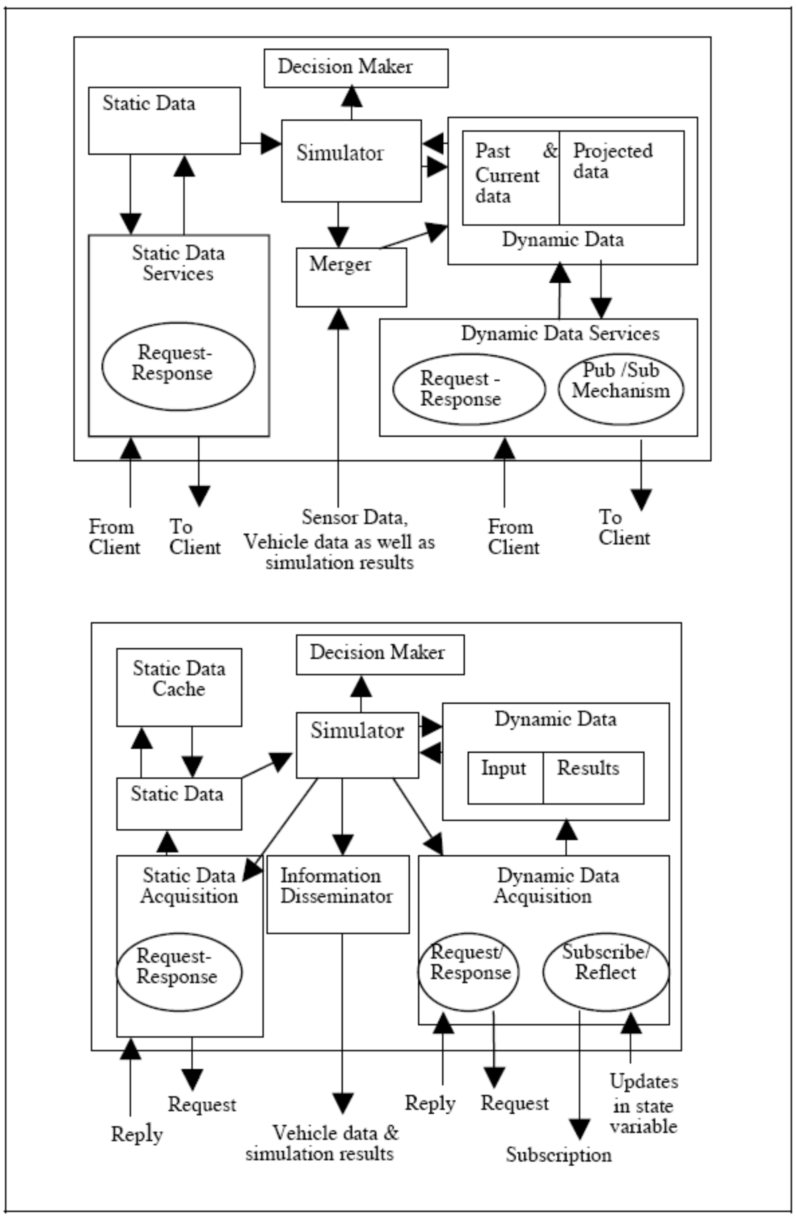


Fig. 2. Software components of (a) server (top) and (b) client (bottom)

The ad hoc distributed simulations proposed here is based on a client-server architecture, where the TMC and the Traffic Controllers act as servers. The major components of both client and software are similar (as is apparent from fig. 2).

In the client system, the static data acquisition unit fetches static data (data that remains unchanged over relatively long periods of time. eg maps of the road network) from traffic controllers. This is stored in the static data cache. The dynamic data acquisition unit fetches data that changes relatively more frequently (vehicle speed, route etc). The dynamic data is fetched from on-board sensors, as well as other nearby sensors. It also has a system for registering for services, in order to receive updates as and when the particular data changes. The simulator module uses the static and dynamic data to make predictions, the results of which are stored in the dynamic data memory area. Decisions arrived at in the decision making nodule typically deal route selection

The server system, has a static and dynamic data services, which can send/receive information from clients through request/response type protocols. Servers have a merger unit (absent in client systems) which *is responsible for resolving inconsistencies and conflicts* [1]. The simulation unit is used for predicting the future state of the system. Like the client system, the decision making module relies on the result of the simulation module to make decisions. For example the decision making unit in a traffic controller may use the simulation results *for signal time optimization* [1].

There are two major distinctions between vehicle based simulations over the and the traditional methods of simulation using fixed sensors. Firstly, vehicle based simulations have to process *a dynamically changing roadway topology* [1], as the vehicle travels over different areas. Therefore through out the simulation the topology information has to be updated. Since traditional simulations get their inputs from fixed sensors, they assume an unvarying topology. Secondly, vehicle based simulations have to be able to *adapt to real time inputs* [1], such as sudden changes in the traffic volume in an emergency. While traditional simulations do take into account emergencies, the unpredictability of human behavior in an emergency would make it difficult to obtain *reliable predictive models* [1] for these situations.

The servers have models that use the raw sensor data and simulation results gathered from the client systems to make macro level analysis. The paper has discussed one such model known as CORSIM. CORSIM (Corridor Simulation) *is a traffic simulation model designed by the Federal Highway Administration for modeling individual vehicle movements on a second-by-second basis for the purpose of assessing the traffic performance of highway and street systems)*[1]. Currently three counties in the Atlanta metropolitan area serve as a test bed. The data required for this model include network geometry (eg number of lanes, free flow speed), traffic control (eg traffic signal) and traffic flow data.

Discussion and Analysis

The paper provides an excellent high level description of a system that uses dynamic data driven simulations to monitor and manage vehicular traffic. The proposed system consists of autonomous objects which are arranged in a hierarchical fashion. Each object carries out simulations pertaining to its area of interest and passes the results onto the next higher level. The client software uses input data and *current values of state variables and is used to calibrate the simulation* [1]. The server uses the *current data from sensors and clients to calibrate simulations and predict the future system states* [1]. This brings out the dynamic data driven characteristic of the simulations.

The server software currently makes use of the CORSIM traffic simulation model for monitoring and assessing vehicular traffic. It is not clear from the paper, which models are used for the simulations carried out by the vehicle mounted computers. The paper suggests the use of adaptive algorithms, similar to those used in the robotics domain. Such algorithms may be used in scenarios where teams of heterogeneous robots need to *cooperate in mapping and exploring an unfamiliar terrain* [1]. The *middleware in both the domains has the common feature, in that it needs to adapt to a dynamic environment, and tolerate failure, to provide a reliable multi-node communication platform to the higher layers* [1]. One possible model is the Recurrent Neural Network (RNN), which can be trained using examples to *map input sequences to output sequences* [2].

The paper does not explain what methods will be used for estimating traffic parameters such as the average traffic speed and density along a given motorway. An algorithm for estimating the parameters, proposed by V. Astarita et al. of the University of Calabria, Italy, suggests the following approach.

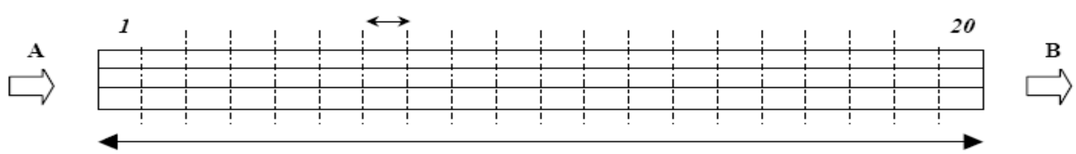


Fig.1 – Test network with a single on ramp (A) and a single off ramp (B).

The algorithm divides the roadway into equal sized cells. Vehicles equipped with wireless radios or mobile phones act as the primary sensors. The number of "instrumented vehicles" entering and exiting a cell are known. Also the number of ordinary vehicles entering and exiting a cell is known by using on and off ramps. The ratio of the total number sensor mounted vehicles to the ordinary vehicles is then used to arrive at an estimation of the average traffic speed , density and flow in each cell in a certain interval of time [3].

Examples of other road traffic management systems.

1 . ARTIMIS (Advanced Regional Traffic Interactive Management & Information

System). The ARTIMIS System is designed to *provide incident, congestion, and freeway management for the Cincinnati-Northern Kentucky Region* [4]. The system uses over 80 CCTV cameras placed on Cincinnati-Northern Kentucky freeways. They cameras send their information to the Control Station via fiber optic and telephone cables.

2. DynaMIT (Developed in MIT). *This real-time traffic management system that combines speed and traffic data generated by video cameras and roadway sensors, with sophisticated simulators that help predict traffic congestion in an area up to an hour into the future* [5]. Much of the technology currently being used in the DynaMIT system is already in use at the Boston Transportation Department's operation control centers.

References

1 - Dynamic Data Driven Application Simulation of Surface Trasportation Systems - R. Fujimoto et al.
2 - "http://www.idsia.ch/~juergen/rnn.html" - Jurgen Schmidhuber's webpage on Recurrent Neural Networks.
3 - Motorway Traffic Management and Traffic Parameter Estimation From Mobile Phone Counts - Vittorio Astarita et al Univeristy of Calabria, Italy.
4 - "http://www.artimis.org" - ARTIMIS webpage.
5 - "http://web.mit.edu/its/pressclips.html" - MIT Press clips webpage on Intelligent Transport Systems.