ECN-based Mitigation of Congestion in Urban Traffic Networks *

Levente Alekszejenkó[1] and Tadeusz Dobrowiecki[1]

Budapest University of Technology and Economics, Budapest, Hungary
ale.levente@gmail.com

Abstract

Traffic congestions cause many environmental, economic and health issues. If we are unable to completely get rid of them, the least we shall try to do is to move them outside of residential areas.

In this paper, a novel signal coordination method is proposed, which aims to mitigate traffic congestions. The proposed algorithm is based on the explicit congestion notification protocol, which is well-known from the domain of computer networking.

Our method was tested under Eclipse SUMO. Results show that the proposed algorithm successfully limits the traffic density and the traffic flow to a certain level.

1 Congestion Problem – A Local Perspective

Nowadays, as the number and the size of the road vehicles are rapidly increasing, even more frequent and longer traffic congestions are formed. Therefore, we might encounter heavy traffic in areas where they can cause even more harm than on main roads or highways.

For example in residential areas or near hospitals vast amount of pollution, noise and vibration, coming from the vehicles, can cause health issues. In areas near nursery and elementary schools they can also pose a safety risk. Moreover, the modern city planning is about to ban vehicles from historical city centers as well. The aforementioned areas usually build up of small, narrow, sometimes even dangerously steep roads with many *right-hand-rule* intersections.

These examples show that it would be really beneficial to avoid heavy traffic to reach specific parts of our cities. Of course, this means there might be areas where the congestions will be even bigger than today, but it might be possible to handle the increased traffic more efficiently there (by e.g., variable speed limits, bi-directional lanes on highways and so on), than in the regions mentioned above.

In this paper, we propose a novel traffic signal coordination method which is capable of restraining heavy traffic from reaching a certain area, which ensures that the density of the vehicles is kept below a critical level, therefore resulting in mitigation of congestion on residential road networks.

2 ECN Protocol and its Adaptation to Urban Traffic Networks

Congestion does not uniquely appear in the domain of vehicle traffic, it is also present in computer networks as well. There are numerous solutions which aim to prevent or handle

*The research reported in this paper was supported by the BME NC TKP2020 grant of NKFIH Hungary, as well as by the BME- Artificial Intelligence FIKP grant of EMMI (BME FIKP-MI/SC).

the congestion in computer networks. Our idea is that some of these algorithms might be applicable to achieve our goal, mentioned above. This conjecture is based on the insight that these protocols prevent packages from being send with too high transmission rate, therefore the network can handle the incoming messages with ease.

Moreover, many other elements are similar in computer networking and the road traffic domain. The networks themselves, for example, are built up of nodes (routers vs. intersections), which might be even capable of actively managing the flow of packets or vehicles in the network. The message packets and the vehicles also form a quantum stream, which contains unique entities with a space between them. The major difference is that vehicles have physical dimensions too, therefore some protocols of computer networks cannot be applied in the domain of road networks.[1]

One of the well-known congestion avoidance algorithms in computer networking is called *Exponential Backoff*. The basic idea behind the exponential backoff is that in case of a collision (which roughly means that a congestion is forming in the network) the transmitters have to wait for a random time between 0 and 2^c μs to resend their messages, where c stands for the number of unsuccessful transmissions. Obviously, this protocol cannot be applied in the domain of road traffic.[2]

Another example of congestion reduction in computer networks is called *Sliding Window Protocol*. This protocol limits the number of packets that can be transmitted at any given time, and as a result it prevents forming congestions. Unfortunately, this protocol might also cut platoons into half, however, platooning is proved to be really beneficial in the traffic domain. Hence, this method is unsuitable for our purposes.

The algorithm which can be easily applied in road traffic and computer network domains as well, is called the *Explicit Congestion Notification* (ECN) protocol [1]. The main idea behind it is that the routers (or in our case the traffic light controllers) can sense somehow the formation of a congestion. If it happens, the routers inform the corresponding transmitters about this fact (let us call these piece of information an *ECN-signal*). If an ECN-signal has been received, the transmission level shall be reduced toward the sender of the ECN-signal.

3 Percepting and Mitigating Congestion

3.1 Overview of ECN-based Traffic Signal Coordination

Intelligent traffic light controllers can be modeled as intelligent agents, which can communicate with each other. Let us call them *judges*, for convenience, and the ones running the ECN-protocol then will be called *ECN-judges*. These communicating judges form a layer in a multi-agent intelligent system (another layers are the communication between vehicle agents and the communication between vehicle platoons and the intersection judges). In our research, we suppose that the communication is free from errors, and the agents themselves are cooperative, trustworthy and bona-fide.

We assume that an ECN-judge can sense somehow the formation of a congestion, and can inform its topological neighbors upstream about this fact by sending out an ECN-signal. If an ECN-signal arrives, the ECN-judge can alter its program accordingly, hence reduces its throughput towards the forming congestion. In any other case, judge should control the traffic

[1] I.e., these protocols use some operations, e.g., dropping or reorder packets, which take advantage of the non-material existence of data.

[2] Most of the people would be really angry, if they were unable to go out from their garages for a long time, because some nearby intersections cannot receive any more vehicles at the moment.

like an actuated traffic light controller, since it has been proved that actuated traffic lights are one of the best ways [2] to optimize the flow of the traffic.

The tricky part is, that the state-space of a single intersection is enormous, regarding the incoming traffic demand, the received ECN-signals and some traditional expectations (e.g. to be fair). Therefore, it seems to be impossible to store an appropriate TLS-program for every situation, see Appendix A. Consequently, signal plans should be generated in real-time. In the following sections, the components of the proposed system are described in detail.

3.2 Sensing the Formation of a Congestion

Detecting the formation of a traffic congestion is a really challenging task. Our research did not focus on this particular problem, therefore we used here a simple solution.

By analyzing the data, which is supported by the loop detectors, the maximum traffic flow can be found on a specific edge. This traffic flow value corresponds to a particular level of occupancy of the given edge.

Above this occupancy level, we suppose the traffic flow will decrease, meaning that a congestion is forming. Therefore, we shall avoid reaching this point, by limiting the occupancy level to 90% of what corresponds to the maximum flow.

In this way, the occupancy limit is set for all edges (traffic lanes) entering into an intersection that is controlled by an ECN-judge. Every now and then (i.e. ca. every 15 s) the ECN-judge calculates the occupancy levels (as the moving average in consecutive time windows) along these edges, and if somewhere the set limit is reached, an ECN-signal with congestion notification is sent out. If the occupancy level falls below the limit, the ECN-signal informs the other judges that the congestion along this particular edge has been dissolved.

3.3 Generating a Signal Plan

ECN judges have to generate signal plans online. These signal plans are based on a simple round robin scheduling, resulting in a fair schedule for all directions. The phase times are adaptively set, and the plans are also influenced by the congestion state of the neighboring intersections, coupled to the intersection which is governed by an ECN-judge.

Computation of a simple signal phase when generating a traffic light system program (TLS-program) can be formalized as an *integer programming problem (IP).* Its goal is to maximize the number of directions which may receive a green light. The matrix of constraints defining the problem is composed from the so-called conflict matrix of the given intersection, describing which directions cannot receive a green light simultaneously, due to the risk of accidents. The other part of the constraint matrix are the so-called additional constraints. Here the logic of the scheduling can be defined as well as the desirable reaction to the incoming ECN messages (i.e., describing which direction need or may not receive a green light at the moment). These components shall be set in accordance to the actual traffic and ECN-notifications.

By solving this IP problem, a signal plan can be obtained.

3.4 Periodic Recalculation of Signal Plans

In order to ensure the periodic working of an ECN-judge, signal plans shall be recalculated every now and then. Let us call the time between two recalculations as phase time (T). This time naturally depends on the number of the vehicles which currently receive a green light (N_v).

Using this parameter the phase time is calculated using equation (1).

$$T = \begin{cases} N_v \cdot 1,5\ s + 5\ s, & \text{if } N_v \leq 23 \\ 40\ s & \text{otherwise} \end{cases} \tag{1}$$

4 Extending Eclipse SUMO

4.1 Previously Developed MAS System

In our previous works [3, 4], a cooperative multi-agent system has already been implemented by extending Eclipse SUMO [5]. This system consists of connected autonomous vehicles, the so-called *smart cars* and intelligent traffic light controllers (*judges*), which, nevertheless, were not connected to each other. The smart cars and the judges were able to communicate with each other. When the smart cars approached an intersection, they requested permission from the corresponding judge to pass through. The judges used simple scheduling algorithm to find out when this permission shall be granted.

In that earlier system, smart cars, which are following closely each other and have exactly the same trajectories, can form groups, so-called *platoons*, before entering an intersection. Such platooning method can somewhat improve the traffic flow by reducing the impact of changing lanes. Another benefit of this method is that only the leader of a platoon needs to exchange messages with the judges, as every other member of a platoon has to follow the vehicle ahead of it. The reduction of the exchanged messages can significantly improve the performance of the system by lowering the computational demand on the side of the judges.

The interface between the intelligent agent system extension and the base core of Eclipse SUMO was provided by the mechanism of a device. Another modified component was the `SL2015` lane change model. This modified LC-model ensures that vehicles, which are forming a platoon, can change lanes together.

As a part of our current research, the ECN-judges were integrated into this ecosystem. Since the earlier system had been created by modifying some of the SUMO's C++ source-code, the new ECN-judges were also implemented by directly using it and the original codes of SUMO.

4.2 Integrating ECN-judges

In the previous system, an abstract class of intelligent judges had already been defined, therefore ECN-judge was implemented as a child of that abstract class. The abstract judge class uses the concept of *conflict classes*. A conflict class is a group of vehicles which can pass through an intersection simultaneously, which means they are equivalent, and can be treated, from a scheduling-theory point of view, as a large single entity.

Unfortunately, conflict classes are not entirely beneficial when TLS-programs are generated in real-time, because we cannot really differentiate vehicles into more than two classes: one class for those vehicles which currently receive a green light, and another class for those which do not receive a green light in a given moment. Hence, an ECN-judge has to change conflict classes of the vehicles when it switches phases, by e.g. removing them from the class which receives a green light and moving them to the class which currently receives a red light. By this method, the ECN-judges can be integrated into our previously proposed multi-agent system (see Figure 1).

One of the most important issues was to obtain the occupancy state of those edges (lanes) which join to the intersection which is controlled by an ECN-judge. To solve this

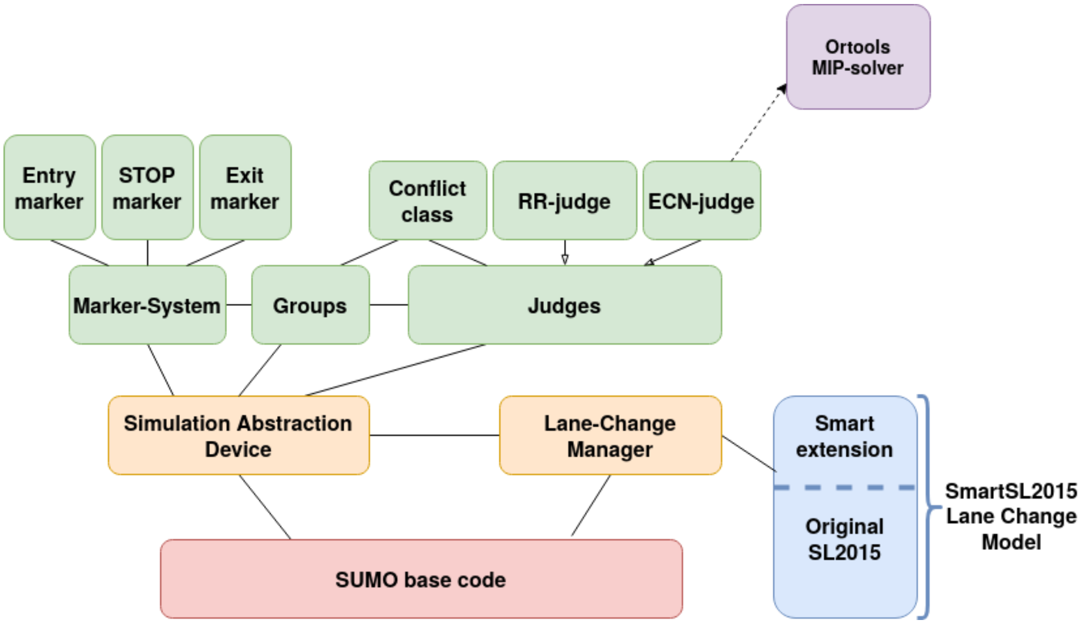


Figure 1: Overview of the extended multi-agent system, based on Eclipse SUMO. SUMO's core functions are interfaced by the two components colored as orange. The parts of the intelligent agent system are colored green. The MIP-solver of Ortools is an external library, developed by Google and capable of solving our integer programming problem.

problem, SUMO's TraCI library was used. This library contains functions (specifically the `libsumo::Edge::getLastStepOccupancy` function) which return the current occupancy state of a given lane. The ECN-judges use this value when calculating whether a congestion is about to form.

The ECN-signals are transmitted as a broadcast message between the judges. As a configuration input, every ECN-judge knows its topological neighbors, therefore when one of its neighbors sends an ECN-signal, indicating a congestion, the phase plans can be changed accordingly. This modified TLS-program will forcibly reduce the throughput towards the forming congestion.

The last problem was to integrate the IP-solver component into the extended Eclipse SUMO-based platform. The used solver is the *OR-Tools Mixed-Integer Programming* toolkit developed by Google. Technically it was simple to add this package, because OR-Tools also use `Cmake` build system. The performance of this toolkit seem to be convincing. As Table 1 shows, our intelligent system can run almost exactly as fast as the original SUMO code.

Simulator	**Scale 1**	**Scale 2**	**Scale 5**	**Scale 10**
Original SUMO	2841.80	1460.32	200.98	77.06
MAS SUMO	2797.78	1367.88	207.66	67.53

Table 1: Comparison of the performance of the original and the extended version of SUMO. Real time factors, provided by SUMO, regarding the scaling of the original traffic demand.

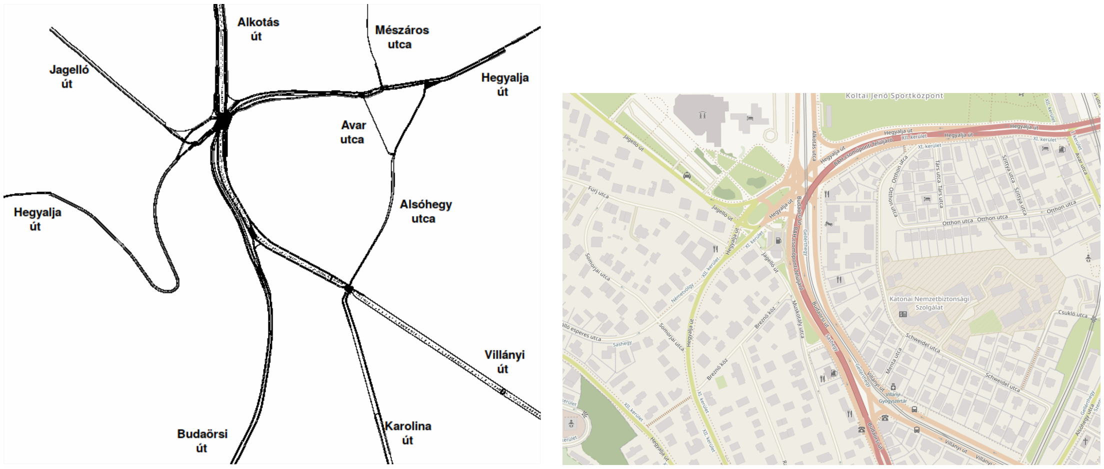


Figure 2: Simulated network of BAH-intersection (left). The central part of the intersection (right) (source: http://osm.org/#map=17/47.486/19.025).

5 Experiments

5.1 The Test-Scenario

Since banning vehicles with old combustion engines from historic city centers is a hot topic nowadays, we applied our system to a major intersection of Budapest, called the BAH-intersection[3] (see Figure 2). To the South of this intersection the M1 and M7 highways terminate, which presumably handle heavy traffic much more easily, than the area to the North from BAH, which is the historical town of Buda, or the residential areas at the eastern and western sides[4] of this intersection. Therefore, our presumption is that applying ECN-judges in this intersection would be beneficial in order to mitigate congestions in the inner-part of the city.

The BAH-intersection and the main roads of its surrounding were fed into Eclipse SUMO [5]. At BAH-intersection, three main roads and two smaller streets intersect at three different junction, topologically close to each other. Moreover, since most of the left turns are prohibited, they also form a bottleneck in the simulated network. For these reasons, ECN-judges shall be placed at these junctions, connected to each other (see Figure 3).

The simulated traffic demand was like a typical workday morning situation (in our measurements, we refer to this case as *Scale 1*[5]). For higher demands, the number of inputted vehicles of this original situation were upscaled by a factor of 2, 3,... 10.

[3] The abbreviation of BAH stands for the three biggest roads which intersect at this point of the city: Budaörsi road, Alkotás street and Hegyalja road.

[4] There is even a natural reserve (Sashegy – Eagle Hill) on the western part of BAH intersection.

[5] As exact values are currently not available from the Road Agency of Budapest, an estimated number of vehicles were used in our simulations. In the morning, the majority of the traffic is coming from the highways which terminate in the Budaörsi road. Significant traffic comes from the Jagelló and Hegyalja roads as well. The most vehicles want to go East on the Hegyalja road, because it drives to one of the bridges over the Danube. Alkotás road is a North-South corridor of the Buda-side, used by about the same amount of vehicles in both directions. About 40% of the traffic in our simulations went on the Hegyalja street Eastbound, and about 20% left the city on Budaörsi street, Southbound. Budaörsi street and Alkotás street (in both directions) handle about another 25% of the traffic. The other 15% of our traffic is randomly distributed among other, not-so-typical routes.

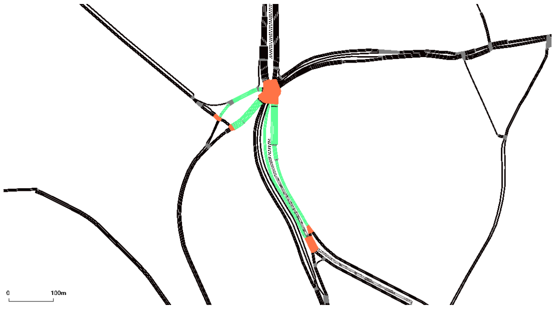


Figure 3: The connected ECN judges. The junctions which are colored as orange are controlled by ECN-judges, which are connected to each other.

5.2 Simulations and their Results

Two different situations were simulated. In the first situation, every traffic light was controlled by an ECN-judge (but only the central three were connected, practically speaking, the others were functioning as simple actuated traffic lights). In the second case, only three central, connected judges were of ECN-type, the others were running a simple Round-Robin scheduler. This allows us to compare the effect of the ECN-judges in itself, instead of comparing the results of the intelligent multi-agent system to the results of the traditional system.

From our previous measurements [3,4], the traffic flow and density values were known, and provide us with a basis for the comparison of the ECN-judge system to the unconnected, yet multi-agent system based solutions (Round Robin (RR) based, practically behaving like an actuated, phase-skipping traffic light) and to the traditional solutions as well. We compared cases when all the traffic controllers were ECN-judges (ECN-only) (much like RR, but instead of scheduling *phases* they schedule *directions*). In the case of ECN-mixed only the three intersection controllers shown on Figure 3 were ECN-type, every other intersections were RR-type.

As the results show (see Table 2 and Figure 4), at low demand levels (at Scale 1 and Scale 2), all of the tested solutions can provide roughly the same results. Then comes a point (around Scale 4), from where the ECN-judges can restrain the density of the traffic, which also means that the traffic flow is limited. As we scaled the traffic demand to higher levels (Scale 8 and Scale 10), throughput of both the traditional and the simple Round-Robin based intelligent solution started to degrade. On the other hand, the ECN-judges were able to stabilize the traffic density and the traffic flow at a certain level, almost regardless of the actual load.

	Scale 1		**Scale 2**		**Scale 4**		**Scale 8**		**Scale 10**	
	D	Q	D	Q	D	Q	D	Q	D	Q
Traditional	22.5	1319	45.0	2617	89.2	4577	137.1	3713	138.4	1098
RR-only	24.4	1267	44.9	2477	NA	NA	139.2	3352	144.7	1287
ECN-only	22.4	1211	44.8	2079	59.3	1987	77.2	2849	78.5	2750
ECN-mixed	22.4	1274	44.9	2120	71.3	2500	83.4	2957	102.8	3782

Table 2: Macroscopic parameters obtained from simulations with different systems. D stands for vehicle density in $[\frac{veh.}{km}]$ and Q stands for traffic flow in $[\frac{veh.}{h}]$ Unfortunately, the simulation with only Round-Robin judges, due to a yet unknown reason, did not provide an output for Scale 4.

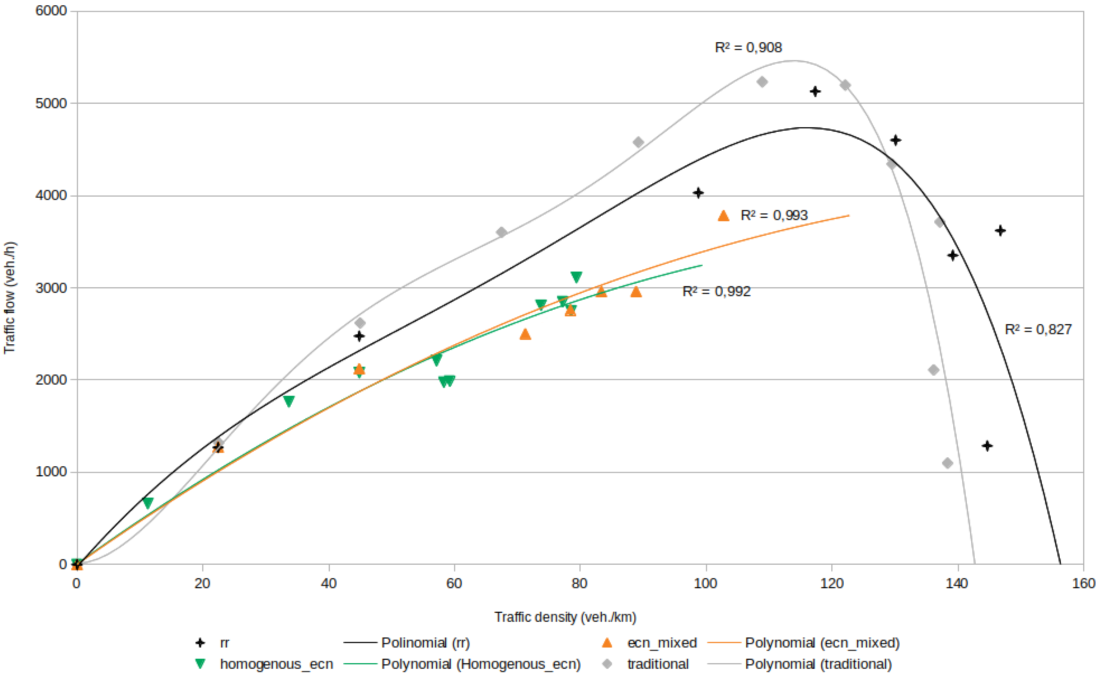


Figure 4: The macroscopic fundamental diagram, provided by the different types of systems. Systems with ECN-judges lack the degrading side of the diagram.

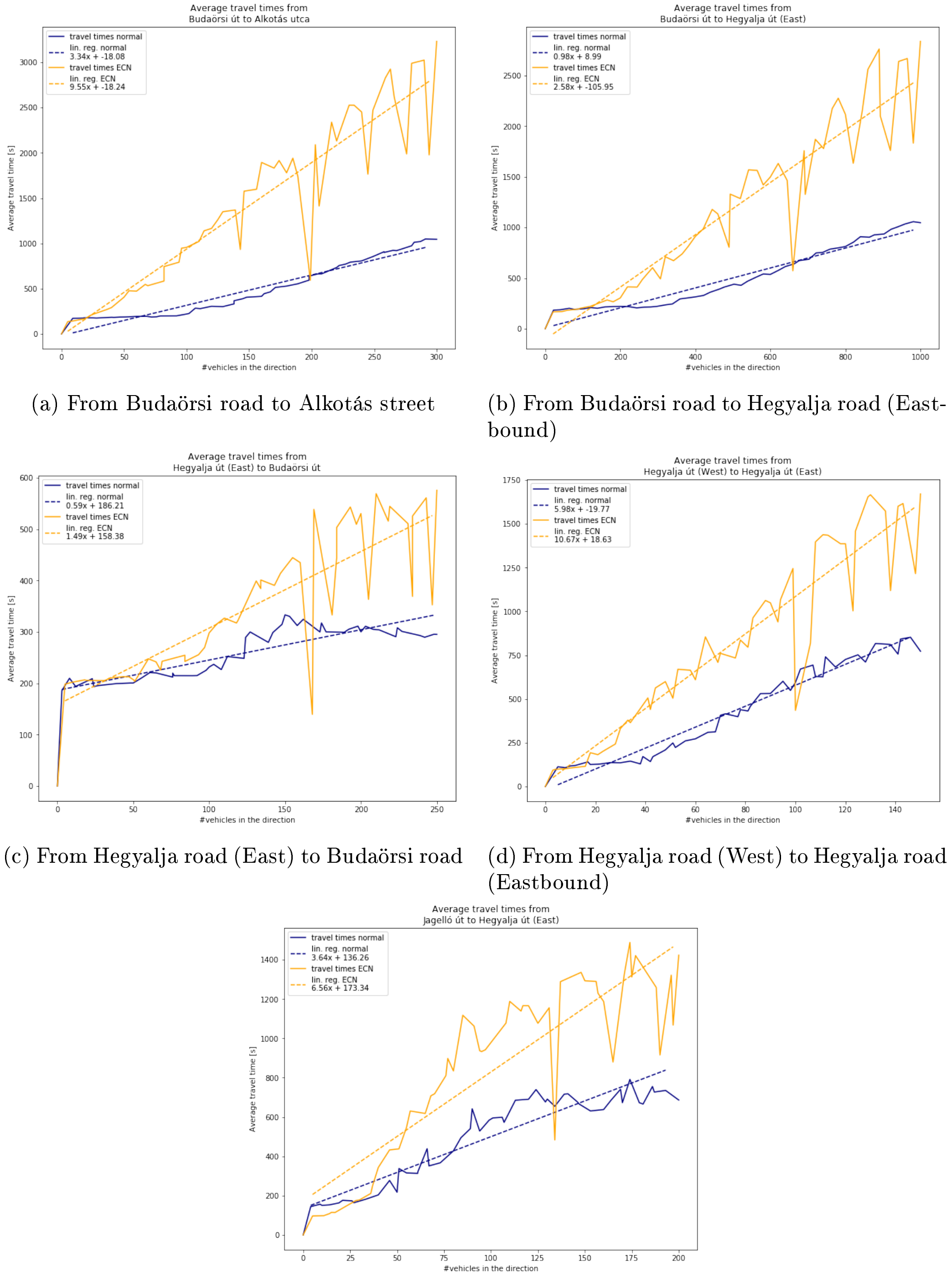
As we found out, that ECN-mixed case behaves slightly better than the ECN-only case, we conducted our measurements with the ECN-mixed setup.

Our expectation might be that the travel time through the congestion-protected edges is significantly increased. This influences the route-decisions of the vehicles, resulting in decreased traffic in this particular area. Thus, on alternative routes, the number of vehicles will be increased accordingly. To analyze the consequences, the average travel times of every vehicle flow were measured as a function of the number of inserted vehicles. As these empirical functions seem to be linear (above a given amount of vehicles), simple linear regressions was fitted to the data points. The equation of these straight line, depicted in Figure 5 can be used to analyze the effects of the ECN judges on the user equilibrium.

Using the `departDelay` and the `departPos` parameters of Eclipse SUMO, the increased probability of congestion on the incoming highways can be assessed[6], see Figure 6. The departure delay somehow reflects how much time it takes to turn on a particular edge (the greater number means more time to wait for a “hole”, which means a denser traffic). Moreover, the departure position reflects how long the congestion gets on the highways (the greater number represents a longer traffic jam). Surprisingly, we can see, the ECN judges create less dense traffic outside of the system perimeter, and in most cases, the length of the congestion is about the same as the traditional traffic control would be used.

Congestions dissolve periodically inside the ECN judge system. When a congestion has just dissolved, the newly incoming vehicles will find really light traffic on the roads, therefore they

[6] For this measurement, length of the Budaörsi street was increased to 5000 m, to model the incoming highways.



(a) From Budaörsi road to Alkotás street

(b) From Budaörsi road to Hegyalja road (Eastbound)

(c) From Hegyalja road (East) to Budaörsi road

(d) From Hegyalja road (West) to Hegyalja road (Eastbound)

(e) From Jagelló road to Budaörsi road (Eastbound)

Figure 5: Travel times and their empirical equations as the function of vehicles traveling in the same direction.

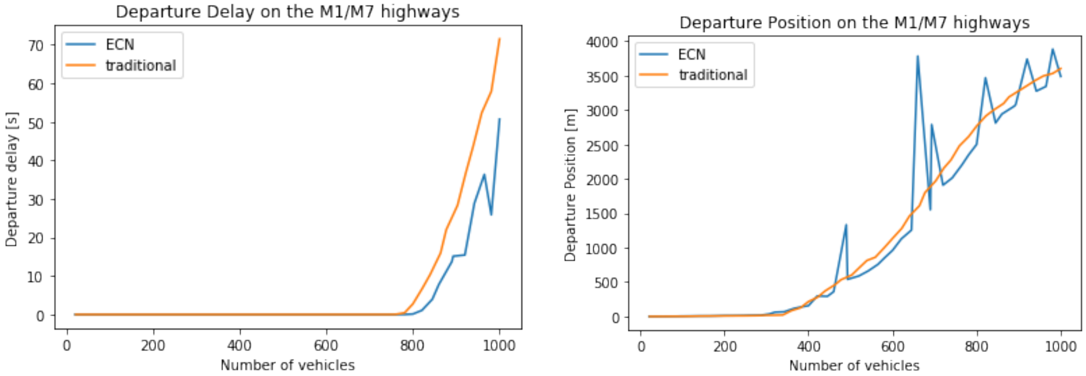


Figure 6: Congestion is increased on the incoming highways. Departure delay (left) is increased on the corresponding edge. Departure position (right) is only moderately increased on the corresponding edge.

can pass through the system more rapidly[7]. This pumping effect allows more distant cars to slowly but constantly move towards the BAH intersection. Perhaps this phenomenon causes on the averagely lighter traffic on the incoming highways.

6 Comparing to Other Approaches

Congestion is an acute problem evident to everybody, no wonder that there are plenty of ideas how to fight it. Some proposals have been tried in practice [6, 7], or only as a theoretical consideration, verified in simplified models by simulations [12, 13, 17–21, 25, 26, 28, 29].

Widely used agent-based paradigm will be even more visible and indispensable in the future, because the character of the possible, or advisable, or demanded level of intelligence for smart cars and their automated decision making capability is not yet clear. With respect to the hierarchically organized multi-agent systems (linking not only ordinary vehicles, but also local intersection managers, then even more centralized area managers, etc.), this paradigm was tried already, see e.g., [8–11, 14, 15, 23, 24, 27, 30].

Results resembling our approach the most are [9,22,24,30]. In those solutions all intersection managers affected by the changes cooperate to adapt signal plans by consensus. In our approach, on the contrary, the affected intersections depend solely on the borders of the sheltered area and on the upstream-downstream direction of the flow. This way the communication load on the agents is less.

The level of an instant occupancy, sensed by detectors, is averaged over a time window, estimated similarly to [9, 24]. In our case the principal novelty is to base the cooperation of the intersection managers on the analogy to the computer network ECN congestion eliminating protocol. At a level of individual intersections signal protocols can be optimized also in a variety of ways, for reviews see [32, 33].

The real vehicle agents may not behave bona fide. Such possibility was considered in [12]. We assume that our agents are well disposed. It is of course a simplification and should be investigated in the future.

[7] Up to the point, when the roads become congested again.

Finally, we did not treat pairing congestion alleviation with route deviation [16,17], or multi-objective problems [31]. We assumed that congestion is the primary problem, and every other problem (related to health, safety and economics) will be almost automatically resolved, once congestions are removed from the city limits.

7 Conclusions

The obtained results can be interpreted that the ECN-judges effectively realize a traffic signal coordination. The aim of this signal coordination, however, differs from the traditional aim of trying to maximize the capacity of the road network. On the contrary, ECN-judges limit the throughput of the network.

Within these constraints, the system can work as a simple controlled road network with actuated traffic lights. If the traffic demand reaches a certain level, this limit will not be exceeded. Naturally, this policy permits to form congestions on the perimeter of the system (i.e. outside of a city), but also ensures that the traffic will be continuous within.

Such reduced traffic would be really beneficial for the residents of a city. The lack of extensive congestions will result in a healthier environment with less pollution, vibration and noise. Road safety will also be increased, therefore bike-riding or riding a scooter would become a more attractive alternative means of transportation.

In the future, a control algorithm shall be developed, which allows to set the traffic limitation to a desired number of vehicles. As far as we know, it strongly depends on the sensing of the congestion-forming.

References

[1] S. Floyd, K. K. Ramakrishnan, and D. L. Black, "The Addition of Explicit Congestion Notification (ECN) to IP." https://tools.ietf.org/html/rfc3168#section-1. last viewed 25. October, 2019

[2] P. Wagner, R. Alms, J. Erdmann, and Y.-P. Flötteröd, "Remarks on Traffic Signal Coordination", in EPiC Series in Computing, 2019, vol. 62, pp. 244–255.

[3] L. Alekszejenkó and T. P. Dobrowiecki, "SUMO Based Platform for Cooperative Intelligent Automotive Agents," in EPiC Series in Computing, 2019, vol. 62, pp. 107–123.

[4] L. Alekszejenkó and T. Dobrowiecki, "Intelligent Vehicles in Urban Traffic – Communication Based Cooperation," in 2019 IEEE 17th World Symposium on Applied Machine Intelligence and Informatics (SAMI), 2019, pp. 299–304.

[5] P. A. Lopez and M. Behrisch and L. Bieker-Walz and J. Erdmann and Y-P. Flötteröd and R. Hilbrich and L. Lüncken and J. Rummel and P. Wagner and E. Wießner, "Microscopic Traffic Simulation using SUMO," in 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2018, pp. 2575–2582., url: https://elib.dlr.de/124092/

[6] How to Fix Congestion, Texas A&M Transp. Inst., https://policy.tti.tamu.edu/congestion/how-to-fix-congestion/, last viewed June 22, 2020

[7] M.D. Meyer, "A Toolbox for Alleviating Traffic Congestion and Enhancing Mobility", Inst. of Transportation Engineers, Washington, DC, 1997, https://rosap.ntl.bts.gov/view/dot/2145, last viewed June 22, 2020

[8] R.Z. Wenkstern, T.L. Steel, O. Daescu, J.H.L. Hansen, and P. Boyraz, "Matisse: A Large-Scale Multi-Agent System for Simulating Traffic Safety Scenarios", Ch 22 in J.H.L. Hansen et al. (eds.), Digital Signal Processing for In-Vehicle Systems and Safety, Springer Science+Business Media, LLC 2012

[9] B. Torabi, R.Z. Wenkstern, and R. Saylor, "A Self-Adaptive Collaborative Multi-Agent based Traffic Signal Timing System", 2018 IEEE Int. Smart Cities Conf. (ISC2), Kansas City, MO, USA, 16-19 Sept. 2018

[10] T. Tettamanti, A. Mohammadi, H. Asadi, I. Varga, "A two-level urban traffic control for autonomous vehicles improve network-wide performance", 20th EURO Working Group on Transp. Meeting, EWGT 2017, 4-6 Sept 2017, Budapest, Hungary, Transportation Research Procedia 27 (2017), pp. 913–920

[11] B. Beak, K. Larry Head, Y. Feng, "Adaptive Coordination Based on Connected Vehicle Technology", Transp. Research Record: J. of the Transp. Research Board, Vol 2619, Issue 1, 2017, pp. 1–12

[12] X. Zhang and D. Wang, "Adaptive Traffic Signal Control Mechanism for Intelligent Transportation Based on a Consortium Blockchain", IEEE Access, Vol 7, 2019. pp. 97281-97295.

[13] Y. Ren, Y. Wang, G. Yu, H. Liu, and L. Xiao, "An Adaptive Signal Control Scheme to Prevent Intersection Traffic Blockage", IEEE Trans. on Intelligent Transp. Systems, Vol. 18, No. 6, June 2017, pp. 1519-1528.

[14] A. Namoun, C.A. Marín, B. Saint Germain, Ni. Mehandjiev, J. Philips, "A Multi-Agent System for Modelling Urban Transport Infrastructure Using Intelligent Traffic Forecasts", in V. Mařík, J.L. Martinez Lastra, P. Skobelev (Eds.): HoloMAS 2013, LNAI 8062, pp. 175–186, Springer-Verlag Berlin Heidelberg 2013

[15] B. Torabi, R. Z. Wenkstern, and M. Al-Zinati, "An Agent-Based Micro-Simulator for ITS", 21st Int. Conf. on Intell. Transp. Systems (ITSC), Maui, Hawaii, USA, Nov 4-7, 2018, pp. 2556-2561

[16] R. Alqurashi and T. Altman, "Hierarchical Agent-Based Modeling for Improved Traffic Routing", 16 Oct 2019, Appl. Sci. 2019, 9, 4376, doi:10.3390/app9204376

[17] J. McBreen, P. Jensen and F. Marchal, "An Agent Based Simulation Model of Traffic Congestion" in Proc. of the 4th Workshop on Agents in Traffic and Transportation, Hakodate, May 2006, 43-49

[18] A. Olia, H. Abdelgawad, B. Abdulhai and S.N. Razavi, "Assessing the Potential Impacts of Connected Vehicles Mobility, Environmental, and Safety Perspectives", J. of Intell. Transp. Systems. Techn., Planning, and Operations, Vol 20, 2016, Issue 3, pp. 229-243

[19] Khan, S. M., "Connected and Automated Vehicles in Urban Transportation Cyber-Physical Systems", PhD Diss., Clemson University, 2019, https://tigerprints.clemson.edu/all_dissertations/2475, last viewed June 22, 2020

[20] X. Ban and W. Li, "Connected Vehicle Based Traffic Signal Optimization", Report C2SMART Center, USDOT Tier 1 Univ. Transportation Center, April 2018.

[21] S.M. Khan, M. Chowdhury, "Connected Vehicle Supported Adaptive Traffic Control for Near-congested Condition in a Mixed Traffic Stream", 14 Jul 2019, arXiv:1907.07243 [eess.SP]

[22] M.-D. Cano, R. Sanchez-Iborra, F. Garcia-Sanchez, A.-J. Garcia-Sanchez, and J. Garcia-Haro, "Coordination and agreement among traffic signal controllers in urban areas", ICTON 2016, Tu.A6.3

[23] M. Abdoos, N. Mozayani, and A.L.C. Bazzan, "Holonic multi-agent system for traffic signals control", Eng. Applications of Artificial Intelligence, Vol 26, Issue 5-6, May, 2013

[24] B. Torabi, R.Z. Wenkstern, R. Saylor, "A Collaborative Agent-Based Traffic Signal System For Highly Dynamic Traffic Conditions", 21st Int. Conf. on Intell. Transp. Systems (ITSC), Maui, Hawaii, USA, Nov 4-7, 2018

[25] H. Nakanishi and T. Namerikawa, "Optimal traffic signal control for alleviation of congestion based on traffic density prediction by model predictive control", Proc. of the SICE Annual Conf. 2016, Tsukuba, Japan, Sept 20-23, 2016, pp. 1273-1278

[26] M. Steingraover, R. Schouten, S. Peelen, E. Nijhuis, and B. Bakker, "Reinforcement Learning of Traffic Light Controllers Adapting to Traffic Congestion", BNAIC 2005, Proc. of the 7th Belgium-Netherlands Conf. on Artificial Intelligence, Brussels, Oct 17-18, 2005

[27] I. Tchappi Haman, V. C. Kamla, S. Galland, J. C. Kamang, "Towards an Multilevel Agent-based

Model for Traffic Simulation", The 6th Int. Workshop on Agent-based Mobility, Traffic and Transp. Models, Methodologies and Applications (ABMTRANS), Procedia Computer Science 109C (2017) 887–892

[28] K. Tumer, Z.T. Welch, and A. Agogino, "Traffic Congestion Management as a Learning Agent Coordination Problem", Multi-Agent Systems for Traffic and Transportation Engineering, 2009

[29] G. Orosz, R. E. Wilson and G. Stépán, "Traffic jams: dynamics and control", Phil. Trans. R. Soc. A 2010 368, 4455-4479

[30] J. France and A.A. Ghorbani, "A multiagent system for optimizing urban traffic", IEEE/WIC Int. Conf. on Intell. Agent Technology, 2003. IAT 2003.

[31] J. Jin, X. Ma, "A multi-objective agent-based approach for road traffic controls: application for adaptive traffic signal systems", Postprint of Paper VI: Jin, J. and Ma, X. (2018). A multi-objective agent-based approach for road traffic controls: application for adaptive traffic signal systems, under review, https://www.diva-portal.org/smash/get/diva2:1205233/FULLTEXT01.pdf, last viewed June 22, 2020

[32] E. Namazi, J. Li, and C. Lu, "Intelligent Intersection Management Systems. Considering Autonomous Vehicles: A Systematic Literature Review", IEEE Access, July 8, 2019, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8756239&tag=1, last viewed January 30, 2020

[33] L. Chen, C. Englund, "Cooperative Intersection Management: A Survey", IEEE Trans. on Intell. Transp. Systems, Vol 17, Issue 2, Feb. 2016, pp. 570-586, DOI:10.1109/tits.2015.2471812

A State Space of an ECN Judge

To understand the difference between the required amount of memory of a simple phase-skipping traffic light and an ECN-based traffic light, let us suppose that we have a grid-like road network, see Figure 7. Every intersection is equipped with the same type of traffic controller. Now, let us focus on a simple junction of this network. This junction is believed to be a plain, four-legged intersection.

If this junction is equipped with phase-skipping traffic lights, at least 4, but at most 8 phases are enough to provide safe passing through for every vehicle. Our actions can be to skip 0, 1, 2, or even 3 phases. This gives us the possibility to choose 0, 1, 2, or 3 phases out of four. Using the well-known formula this means that we have $\binom{4}{0} + \binom{4}{1} + \binom{4}{2} + \binom{4}{3} = 15$ choices. Generally, Equation (2) gives the number of choices, which an n-legged intersection can provide. Note that the number of states, in this case, is proportional to 2^n.

$$\sum_{k=0}^{n-1} \binom{n}{k} = 2^n - 1 \tag{2}$$

Now, let us consider the case of an ECN judge. As all traffic controllers in the road network are ECN-based, every *direction* can be restricted. However, it does not necessarily mean that a phase shall be skipped. It is much more alike a supplementary green light of a traditional system, which allows turning right, even in the time when the main lamp shows red. Therefore, not only the phases but all the directions have to be represented in the state space.

Directions can be modeled as a (directed) graph of which nodes are the incoming and outgoing streets. The edges of the graph represent the connections between every street. As a complete graph, with n nodes, has $\frac{n(n-1)}{2}$ edges, we have the option to choose directions "randomly" from this amount of possibilities. Analogously to the case of a phase-skipping traffic light, Equation (3) shows the required size of the state space. As we can see, it is

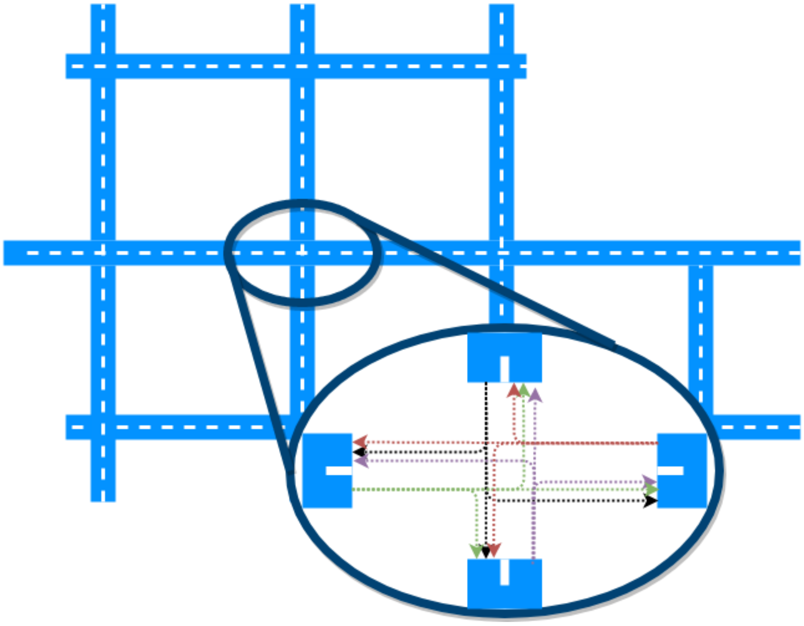


Figure 7: ECN judges controls directions, instead of phases. It means, storing its program requires exponentially more memory than a simple phase-skipping program.

proportional to $2^{\frac{n(n-1)}{2}}$ which is an exponential growth compared to the traditional case.

$$\sum_{k=0}^{n-1} \binom{\frac{n(n-1)}{2}}{k} = 2^{\frac{n(n-1)}{2}} - 1 \tag{3}$$

The controlling algorithm is likely to be realized by two components. One component is a *Look-Up Table, LUT*, and the other one is a component that searches in this LUT for proper configuration of the "traffic lamps". Searching itself can be implemented powerfully, but the size of the LUT cannot be smaller than the actual size of the state space. It means, even a four-legged intersection would require a 63-sized LUT (compared to a phase-skipping controller's requirement of a 15-sized LUT). If we have a greater intersection, with five legs, these numbers will be 1023 and 31, respectively. An even bigger, six-legged intersection with ECN judges will require a LUT capable of storing 32767 entries, meanwhile, a phase-skipping system would use only 63 entries.