Possibilities and Limitations of Decentralised Traffic Control Systems

Sven Tomforde, Holger Prothmann, Jürgen Branke, Jörg Hähner, Christian Müller-Schloer, and Hartmut Schmeck

Abstract—Due to steadily increasing mobility and the resulting rising traffic demands, serious congestion problems can be observed in many cities. One promising approach to alleviate the congestion effects is the coordination of the network’s traffic signals in response to the traffic flow. The recently introduced Decentralised Progressive Signal Systems approach is an adaptive coordination mechanism for traffic signals in urban road networks that relies on local traffic data only. Since the decentralised process cannot lead to optimal results in some special cases, it is extended with an optional hierarchical component introduced in this paper. Based on a broader view on the current network traffic, this Regional Manager is responsible for determining which intersections are coordinated. The efficiency of the coordination determined by the Regional Manager is demonstrated in a simulation-based evaluation that considers the decentralised mechanism and an uncoordinated system for comparison.

I. INTRODUCTION

Urban road networks suffer from serious congestion problems due to constantly rising traffic demands. For urban areas in the US, the Urban Mobility Report [1] estimates that about 4.2 billion hours of delay resulted from congestion in 2007, leading to an additional fuel consumption of 10.6 billion litres. Expressed monetarily, congestion has caused follow-up costs of $87.2 billion, which is an increase of more than 50% over the previous decade.

Strategies to satisfy the rising demands are manifold, including the construction of additional roads, the strengthening of public transportation, or a better distribution of trips during the day. Additionally, it is important to get as much service as possible from the available infrastructure. To this end, the efficient control of traffic lights is an important factor.

The task of efficiently controlling and coordinating traffic signals in urban road networks is, however, highly complex and requires continuous adaptation to changing traffic demands. One approach to deal with this type of problem is Organic Computing [2], a research field that aims at coping with complexity by building adaptive and self-organising systems that distribute a global task among autonomous entities.

In previous work [3], [4], the generic observer/controller architecture used in Organic Computing [5] has been applied to adaptive traffic control in urban road networks. The resulting traffic control system provides a decentralised mechanism to coordinate adaptive and learning intersection controllers to form progressive signal systems (PSSs, sometimes also called “green wave”). The approach has been shown to reduce the delay and the number of stops in urban road networks. However, as the decentralised mechanism relies on locally available knowledge only, it leads to a suboptimal coordination of intersections for some traffic demands. To handle these cases, an additional hierarchical observer/controller component – the Regional Manager – is proposed in this paper. The Regional Manager supports the coordination by collecting data on the current traffic demands from the intersection controllers and by determining which intersections should form PSSs. It therefore provides an additional extension to the decentralised system that can further improve the system performance. It is important to note that the Regional Manager is not meant to take over central control. It supports the local decisions taken by the decentralised traffic light controllers but cannot become a bottleneck.

The remainder of the paper is structured as follows: Section II briefly discusses the coordination of traffic lights in road networks and reviews existing adaptive network control systems. Section III recapitulates the decentralised coordination mechanism, while the Regional Manager is motivated and explained in Section IV. The resulting hierarchical control system is evaluated in a simulation study. Results of the simulations are presented in Section V. Finally, Section VI concludes with a summary and an outlook.

II. STATE OF THE ART

The control of traffic signals at an intersection typically involves four basic signalisation parameters: cycle time, split, phase sequence, and offset. The cycle time determines how long it takes until all signal phases have been activated and the cycle is restarted. The sequence of all phases to be considered is used to split this cycle into fractions of appropriate lengths. Finally, if the intersection can coordinate its operation with neighbouring intersections to generate a PSS, a global timer is used and the starting point of the cycle is shifted by a certain offset. The combination of the basic signalisation parameters leads to a signal plan for the intersection.
Due to the importance of efficient traffic control, adaptive network control systems that control the basic signalisation parameters in response to current traffic demands have been developed for many years. The systems model the traffic flows in a network and evaluate possible signal control alternatives on-line. This section presents the adaptive network control systems SCOOT, OPAC, and MOTION as examples of centralised, decentralised, and hierarchical network control and emphasises their differences from the organic approach presented here. More extensive overviews are available in the literature [6], [7].

A. SCOOT

The **Split, Cycle, and Offset Optimisation Technique** (SCOOT, [8]) is one of the first adaptive network control systems that was successfully applied in the field. SCOOT centrally computes a single cycle time for all intersections in the network and adjusts splits and offsets for the individual intersections in order to reduce delays and stops. For split adjustment, SCOOT investigates whether shortening or extending a phase reduces the degree of saturation for the most heavily saturated approach and accordingly adapts the split in small steps. Similarly, the cycle time is slightly adjusted every few minutes aiming to achieve a degree of saturation of approximately 90% for the most heavily saturated intersection. Finally, offsets are adapted once per cycle using performance measurements derived from a simple traffic model. Again, offset adaptation is performed in steps of a few seconds only.

The benefits obtainable by SCOOT installations are documented in several studies (e.g., [9]). However, SCOOT is also criticised for the use of fixed phase sequences and its stepwise change of signalisation parameters that results in a relatively slow adaptation process [10].

B. OPAC

The **Optimisation Policies for Adaptive Control** (OPAC, [11]) have been developed to support the traffic-responsive control of single intersections. In contrast to most other adaptive network control systems, OPAC works decentralised in the sense that its control algorithm is applied independently at each intersection and without an explicit coordination.

OPAC relies on dynamic programming to compute a local switching policy that is optimal for a short time horizon in the future. Using a rolling horizon approach, the derived policies are continuously applied and recalculated in short intervals. However, some adaptations are required in practice [11]: OPAC’s policy calculation needs detailed vehicle arrival data for the entire horizon. Usually, such data cannot be obtained from available detection systems, therefore a prognosis model is needed to derive the missing data. Furthermore, dynamic programming requires an extensive computational effort, so a faster heuristic replaced the dynamic programming algorithm that was initially used.

OPAC can offer remarkable benefits compared to traffic-actuated controllers [12], but has been criticised for the missing explicit coordination among the intersections [10].

C. MOTION

The **Method for the Optimisation of Traffic Signals in Online-Controlled Networks** (MOTION, [13]) combines centralised optimisation with local traffic-actuated control: On its central level, MOTION determines the most relevant traffic streams in the network and calculates appropriate PSSs. The resulting signalisation constraints are communicated to the intersections in form of frame signal plans. The plans ensure the coordination of intersections within the network, but leave some freedom for local traffic-actuated adaptations.

In detail, signal plans are adapted by MOTION in a four step process: The first step is dedicated to traffic detection. Data on the current traffic demand is gathered and preprocessed for later use. In a second step, a model of the current network traffic is obtained from this data. Missing data is completed, incidents and congestions are detected, the future traffic development is predicted, and the most relevant traffic streams are identified. The third step is then dedicated to the optimisation of control parameters. At first, the network-wide cycle time and phase splits are determined based on the intersections’ degree of saturation. Afterwards, MOTION focuses on the coordination of intersections by optimising phase sequences and offsets based on the most relevant traffic streams with the objective to minimise delays and stops. In a last step, MOTION decides whether the current signalisation needs to be adapted. If the newly obtained signalisation results in a significant improvement, the new signal plans are communicated to the local controllers.

These four steps are repeated in intervals of a few minutes to achieve an on-line adaptation of the signalisation to the current traffic demand. MOTION has been evaluated in several studies (e.g., [13]).

D. Differences from the organic approach

The organic approach proposed here differs in various aspects from SCOOT, OPAC, and MOTION: In contrast to SCOOT, it is not strictly centralised, i.e., it does not solely rely on a control centre for deciding on the network’s signalisation. Strictly centralised systems require an extensive (and therefore expensive) communication infrastructure that is potentially susceptible to failures.

In contrast to the uncoordinated OPAC system, both the decentralised organic approach (Section III) and its hierarchical extension (Section IV) emphasise the importance of coordination for the traffic-responsive creation of PSSs.

Most similarities exist between MOTION and the hierarchical approach presented here: A centralised coordination is combined with local traffic-responsive control. A difference is in the local controllers that optimise their signal plans, while MOTION supports traffic-actuated operations only.

III. DECENTRALISED COORDINATION OF ORGANIC INTERSECTIONS

This paper extends a decentralised mechanism for the coordination of organic intersections with an additional component that supports the coordination process. In the
following, the working principle of the organic intersections and their decentralised coordination are briefly recapitulated.

A. Organic Intersections

To obtain an adaptive intersection with learning capabilities, a commonly used fixed-time or vehicle-actuated traffic light controller can be extended by an observer/controller component [5] that continuously optimises its control parameters depending on current traffic demands (see Figure 1).

The observer/controller is split into two layers: Layer 1 is responsible for the traffic-dependent on-line selection of signal plans. An observer monitors the intersection’s traffic flows and provides the processed flow data regularly to a Learning Classifier System (LCS). An LCS [14] is a rule-based reinforcement learning mechanism that keeps track of a set of condition-action-prediction rules. Here, the rules map traffic situations to appropriate signal plans. Their condition specifies situations for which a signal plan (stored in the action part) can be applied, while the prediction gives an estimate on the expected quality (in terms of the resulting average delay). The signal plan selection is based on the available rules and their predictions, which are updated after each rule activation based on the delays observed at the intersection.

Layer 2 performs off-line signal plan optimisations based on microscopic traffic simulations. An Evolutionary Algorithm (EA) searches for good signal plans for a specified traffic situation and provides the resulting rules to the LCS in Layer 1, so they can be applied at the intersection. This encapsulated rule generation helps avoiding the application of low performing or unsafe rules at the real intersection, since only tested rules are made available to Layer 1.

The resulting architecture is adaptive and can learn control strategies for unforeseen traffic demands, therefore we call it organic. Similar to OPAC, it is meant to provide adaptive traffic control for a single intersection only. To achieve an explicit coordination among neighbouring intersections, a decentralised collaboration mechanism has been proposed that is briefly recapitulated in the following. A detailed description of the organic intersection architecture and the collaboration mechanism is available in [3], [4].

B. Decentralised Coordination

The decentralised coordination mechanism – called DPSS for Decentralised Progressive Signal Systems – is a three step process: In a first step, the intersections determine partners that collaborate to form a PSS. Once the partnerships are established, the collaborating intersections agree on a common cycle time which is a prerequisite for coordination. In a third step, the partners select signal plans that respect the common cycle time, calculate offsets, and finally establish a coordinated operation. The three steps are briefly explained in the following:

Step D.1 – Partnerships To determine a sequence of intersections that can establish a PSS improving the network’s traffic flows, each intersection \( j \) determines which of its local turnings exhibits the strongest vehicle flow. Assuming that intersection \( j \) determines the turning from upstream intersection \( i \) to downstream intersection \( k \) as its strongest turning movement, intersection \( j \) informs its desired predecessor \( i \) that it would like to be \( i \)'s successor in a PSS. After all intersections informed their desired predecessor, a local matching takes place: Each intersection \( j \) checks whether it was chosen by its downstream intersection \( k \) as \( k \)'s desired predecessor. If this is the case, \( j \) acknowledges the partnership with \( k \). Furthermore, reject messages are sent to other intersections that have selected \( j \) as desired predecessor, so no partnership is established with these intersections initially. Intersections that were not integrated in a PSS can repeat the above process with other less heavily used turning movements. When the first step is completed, intersections collaborating in a PSS know their partners.

Step D.2 – Cycle time To determine a common cycle time that is suitable for all intersections in the PSS, the collaborating intersections determine their currently desired cycle length using their observer/controller. Afterwards, an echo algorithm starting at the first intersection of the PSS is used to determine the maximum of the desired cycle times and to distribute this information among the intersections. The maximum is selected since shorter cycles would reduce the capacity of the most heavily used intersection and would therefore result in rapidly increasing delays.

Step D.3 – Signal plans and offsets Finally, signal plans need to be selected and offsets have to be calculated. The process again starts at the first intersection of the PSS which uses its observer/controller to select a signal plan that is suitable for the current traffic demand and respects the common cycle time. Afterwards, the intersection communicates the relevant timing data for offset calculation to its successor. At the successor intersection, the process is repeated until the last intersection in the PSS is reached.

When Step D.3 is completed, the PSS is established. The coordination completely relies on locally available traffic data and on communication among neighbouring intersections,
details are available in [3], [4]. While a decentralised approach has many advantages (like reduced effort and costs compared to centralised systems), there are also drawbacks. The use of local information only can lead to a suboptimal coordination, a problem discussed in the following section.

IV. Hierarchical Progressive Signal Systems

This section extends the DPSS mechanism with an additional hierarchical component – the Regional Manager (RM). The RM is responsible for dealing with special cases within traffic networks where the decentralised mechanism creates a suboptimal coordination. The approach replaces Step D.1 of the DPSS mechanism by aggregating local knowledge to obtain a more global view on the network. Based on this, the RM decides on the creation of PSSs within the network and reuses the decentralised Steps D.2 and D.3 to finally establish the coordination.

A. Motivation

In the DPSS mechanism, the local selection of partners for the PSSs is based on the strongest turning movements – leading to a preferential treatment of the strongest streams within the network. In most cases, this is a useful approach, but it does not always lead to the best possible solution.

The network depicted in Figure 2 introduces an example with sub-optimal performance of the DPSS mechanism. The figure shows a Manhattan-type network with six intersections. In the network, two traffic streams run from west to east, while three streams are running from north to south. Traffic for the other directions is neglected in this example. The width of the arrows is proportional to the traffic flow for the particular stream. The strongest stream is the central of the southbound streams, but in sum both eastbound streams are larger than the three streams running from north to south. The DPSS mechanism initially establishes a PSS for the strongest stream. In further iterations of Step D.1, signal systems are also established for the other southbound streams. However, a better coordination could be obtained by coordinating the two eastbound streams. This becomes clearer when evaluating the number of vehicles that benefit from coordination.

![Traffic flows in a Manhattan-type network](image)

A metric is needed to evaluate the potential benefit of a PSS. The target of the RM is to reduce the average number of stops for all vehicles passing the controlled network by coordinating the network’s intersections. A vehicle driving on a coordinated stream can pass the coordinated intersections without having to stop at a red light. At the first intersection of a PSS, however, there is no benefit from coordination, since the vehicles arrive randomly. Consequently, the benefit of coordination is obtained by summing up the vehicle flows for the coordinated turning movements at all intersections except for the first one.

B. Regional Manager

The algorithm proposed for the RM is a heuristic that aims at finding the best possible combination of PSSs for the network. It works in three steps (see Figure 3): Initially, the local traffic demands at the intersections are aggregated to obtain a regional view on the network. A graph model represents the region’s traffic flows and forms the basis of coordination decisions for the RM. The availability of regional knowledge is an important difference from the previously proposed DPSS mechanism [3], [4] that relies purely on local demands.

In a second step, the region’s traffic flows are analysed and strong traffic streams that serve as candidates for the installation of PSSs are identified. Unfortunately, PSSs cannot be installed for every candidate stream. Establishing PSSs for both directions of an arterial road is, e.g., possible only in special cases. Therefore, a third step is required. The RM combines non-conflicting streams to stream systems and selects the system promising the best benefit for installation in the network. All three steps are depicted in Figure 3 and will be discussed in more detail in the following:

Step RM.1 – Build network graph

The target of this step is to obtain a graph representation of the current traffic flows within the network. A subgraph is created for each intersection representing its topology and the local traffic flows. The subgraph contains one vertex for each incoming section, and one edge for each turning movement. Edges are directed and weighted, with the weight corresponding to the current traffic flow of the turning. The resulting subgraphs are transmitted to the RM, which connects all subgraphs to obtain a network-wide representation of the current traffic demand. Figure 3(a) depicts the graph obtained by the RM for a Manhattan-type network of six nodes like the one shown in Figure 2 (edge weights are omitted).

Step RM.2 – Determine traffic streams

The second step aims at identifying traffic streams for which a coordination is beneficial for the network. To determine the candidate streams, Algorithm 1 iteratively connects graph edges to streams until a stop criterion is reached. Using the graph $G = (V,E)$ created in Step RM.1 (see Figure 3(a)), the RM iteratively builds streams until all edges have been removed from $E$ or the remaining edges’ weights are below a predefined threshold. Therefore, it chooses the edges with the highest weight (Line 3) and iteratively determines the best predecessors (Lines 6–11) and successors (Line 12) by selecting the edges with highest weights from the particular
Algorithm 1 Determine traffic streams

**Input:** A set $E$ of edges (Step RM.1)
**Output:** A set $S$ of streams

1: Sort $E$ w.r.t. to edge weights.
2: repeat
3: Choose edge $e = \text{arg max}_{e \in E} \text{weight}(E)$.
4: Remove $e$ from the set $E$.
5: Create a new empty stream $s$, add $e$ to $s$.
6: Set edge $e^* = e$.
7: repeat
8: Choose predeccessing edge $p$ of $e^*$ with $p = \text{arg max}_{p \in \text{Predecessors of } e^*} \text{weight}(E)$.
9: Add $p$ to stream $s$.
10: Set $e^* = p$.
11: until $e^*$'s intersection contains an edge $e^{**}$ with $e^{**}.weight() > e^*.weight()$.
12: Repeat Lines 6–11 for the subsequent edges of $e$.
13: Remove all edges of stream $s$ from $E$.
14: Set $s\.weight()$ to the number of benefiting cars.
15: Add stream $s$ to set of streams $S$.
16: until $E$ is empty or $e\.weight() \leq$ threshold $t$.

Algorithm 1 has a complexity of $O(m \log(m))$ with $m$ being the number of edges. The time is required for sorting (Line 1). During the execution of Lines 2–16, every edge is looked at a constant number of times only. Edges included in a stream are removed (Lines 4 and 13) and will not be considered in later iterations. The repeated execution of Lines 6–11 and 12 can be performed in linear time using a suitable data structure, since the degree of vertices is bounded by a constant when representing intersections. Furthermore, since each edge is part of at most one stream, evaluating the benefits of all constructed streams (Line 14) can be done in linear time.

The complexity of Algorithm 1 can be denoted in the same way depending on the number of intersections, since the number of turnings – which are represented as edges in the graph – depends linearly on the number of intersections in real traffic networks.

The result of Step RM.2 is a set $S$ of streams with assigned benefits that serves as input for the third step of the process.

**Step RM.3 – Determine stream systems** Finally, the best possible combination of streams is determined. The selected streams need to be non-conflicting, i.e., they must not intersect each other or run in different directions on the same roads. In other words, each intersection can be part of one PSS only. Furthermore, the selected streams should maximise the number of benefiting vehicles, i.e., the sum of the stream evaluations should be maximal. Figure 3(c) depicts several systems of non-conflicting streams for the example network. Streams are again visualised as connected sequences of thick edges.

Algorithm 2 presents the Greedy approach used by the RM to create promising stream systems without generating the power set of streams. In a preprocessing step (Lines 1–8), pairs of conflicting streams are identified. Using the resulting function $J$, stream systems are iteratively created by including the best non-conflicting and unprocessed traffic stream from the set $S$ generated in Step RM.2. Initially, $S$ is sorted and copied to a new set $U$ (Lines 9–10) to assure that each stream in $S$ is processed at least once. The process terminates as soon as $U$ is empty (Line 29). In every iteration of the outer loop (Lines 11–29), the set $T$ contains streams that are not in conflict with any stream in the stream system $Z$. In Lines 13–19, $Z$ is initialised with the best unprocessed stream, conflicting streams are removed from $T$. Afterwards, the inner loop in Lines 20–27 is processed. The loop iterates over the set $T$, adds the best stream to $Z$, and updates $U$ and $T$.

Using suitable data structures, Algorithm 2 has a complexity of $O(m^2)$, with $m$ being equal to the number of edges. The preprocessing (Lines 1–8) needs time $O(m^2)$, since the number $r$ of streams containing the same intersection is bounded by a constant (the intersection's number of turnings). Furthermore, at most $O(m)$ entries of the function...


Algorithm 2 Determine stream systems

**Input:** A set $S$ of streams determined during Step RM.2

**Output:** A set $R$ of generated stream systems

1: Construct a function $J: S \times S \rightarrow \{0, 1\}$.
2: Set $J$ to 0 for all arguments.
3: **for all** intersections $i$ **do**
4:   Determine the streams $s_{i,1}, \ldots, s_{i,r}$ containing $i$.
5: **for all** $j, k \in \{1, \ldots, r\}$ **do**
6:   Set $J(s_{i,j}, s_{i,k}) = 1$.
7: **end for**
8: **end for**
9: Sort $S$ w.r.t. the number of benefitting vehicles.
10: Set $U = S$.
11: **repeat**
12:   Set $T = S$.
13:   Create an empty stream system $Z$.
14:   Choose stream $s_1$ with $s_1 = \arg\max_{s \in U} \text{weight}(U)$.
15:   Add $s_1$ to $Z$.
16:   Remove $s_1$ from $U$.
17: **for all** streams $s'$ with $J(s_1, s') = 1$ **do**
18:   Remove $s'$ from $T$.
19: **end for**
20: **repeat**
21:   Choose stream $s_2$ with $s_2 = \arg\max_{s \in T} \text{weight}(T)$.
22:   Add $s_2$ to $Z$.
23:   Remove $s_2$ from $U$.
24: **for all** streams $s'$ with $J(s_2, s') = 1$ **do**
25:   Remove $s'$ from $T$.
26: **end for**
27: **until** $T$ is empty.
28: Add $Z$ to $R$.
29: **until** $U$ is empty.

$J$ are set to 1 during preprocessing. Therefore, the amortised complexity of the inner loop in Lines 20–27 is $O(m)$, and the loop in Lines 11–29 can be executed in time $O(m^2)$.

The result of Algorithm 2 is a set $R$ of stream systems (see Figure 3(c)). To determine the benefitting vehicles for a stream system in $R$, the previously defined evaluation function is summed up for each contained stream. Based on the obtained values, the best stream system is selected for implementation in the road network. The RM communicates the resulting partnerships to the intersections where a common cycle time and appropriate signal plans and offsets for the PSSs are determined locally using Steps D.2 and D.3 of the DPSS mechanism.

V. Evaluation

To study the possibilities and limitations of decentralised traffic control systems and to evaluate the potential benefits of a hierarchical architecture that has access to network-wide traffic demands, the DPSS mechanism is compared to the hierarchical traffic control system presented in the previous section. In the following, the experimental setup of the simulation study is presented and the obtained results are discussed.

A. Experimental Setup

The investigated test case is a Manhattan-type network consisting of six intersections (see Figure 4). The intersections have an identical topology that allows for all possible turning movements. The connecting road segments are one-laned, have a length of 250 m, and provide an additional side-lane for left-turns when approaching an intersection. Each intersection is equipped with an observer/controller (see Section III-A) responsible for the traffic-responsive reconfiguration of a four-phased fixed-time controller (see Figure 4).

![Simulation model and signal phases](image)

The simulated traffic demand resembles the example discussed in Section IV-A, leading to a suboptimal coordination when using the DPSS mechanism. The traffic demands are listed in Table I for all origin/destination (O/D) pairs in the network. During the first half of the simulation – which lasts for four simulated hours in total – the most heavily used traffic stream in the network starts at $G$ and ends at $H$. Therefore, the DPSS mechanism will create a PSS serving this stream. Furthermore, the signalisation for the other north-south streams will be coordinated by the decentralised mechanism. This coordination, however, is less efficient than creating PSSs for the west-east streams from $A$ and $C$ to $B$ and $D$, respectively. These signal systems would be beneficial for a larger number of vehicles, but this is difficult to detect using locally available traffic data only.

For the second half of the simulation, the network’s traffic demands change their directions, i.e., the most heavily used traffic stream starts at $H$ and ends at $G$, while the most beneficial PSSs coordinate the east-west streams from $B$ and $D$ to $A$ and $C$, respectively. Therefore, the coordination mechanisms need to detect the changed traffic demand and have to adapt the PSSs in the network.

Using the simulation model, the DPSS mechanism is compared to the novel hierarchical approach discussed in Section IV-B, while the uncoordinated organic intersections
(as explained in Section III-A) serve as reference scenario. The comparison is based on

- the resulting travel time and number of stops (network-wide and for selected traffic streams),
- the resulting fuel consumption, and
- the emission of relevant pollutants.

Simulations were conducted using the microscopic traffic simulation software AIMSUN 5.1 [15] running on a 2.5 GHz quad-core machine. The setup of the organic intersections was the same as in previous studies (see [3], [4]). The rule set of the LCS was empty at the beginning of the simulations, i.e., all signal plans were learned on-line using the two-layered architecture depicted in Figure 1. PSSs were updated every ten simulated minutes depending on the current traffic situation. In the following, the evaluation results (averaged over five independent simulations) are presented.

### B. Simulation Results

Figure 5 depicts the network-wide travel times and stops obtained from the simulations. It can be observed that both coordination approaches are able to reduce the number of stops in the network for (nearly) the complete simulated duration compared to an uncoordinated operation. After an initial learning phase that is needed to populate the rule sets of the intersections’ LCSs, an average reduction of 8.2% and 18.8% is obtained for the decentralised and the hierarchical approach, respectively. As expected for the investigated test case, the hierarchical approach achieves a higher reduction of stops, but the DPSS mechanism still performs reasonably well. The abrupt change in the traffic demand (indicated by the dotted vertical line) results in temporarily increased travel times and stops for both approaches until the change is detected and the coordination is adapted.

With respect to travel times no significant reductions were obtained by coordination (see Table II for details). This is due to the fact that coordination leads to increased cycle times for all but the most heavily used intersections within a PSS (see Step D.2 for details on cycle time determination). The increased cycle time of the less utilised intersections results in increased local delays whenever the intersections cannot benefit from coordination (i.e., when an intersection is the start of a PSS). These local delays contribute to the network-wide travel time.

To illustrate the differences between decentralised and hierarchical coordination, two selected streams have been evaluated with respect to travel times and stops. Figure 6(a) depicts the results obtained for Stream $G \rightarrow H$ which is the most heavily used stream during the first half of the simulation. During this period, the DPSS approach creates a PSS for this stream that leads to decreased travel times and stops. After the traffic demand changes at the beginning of the second half of the simulated period, the stream is no longer coordinated and as a result travel times and stops rise to a level similar to the one obtained for uncoordinated operation. Nevertheless, a reduction of 11.7% with respect to travel time and 22.0% with respect to stops is obtained for this traffic stream by decentralised coordination when the whole simulated period is considered.

Figure 6(b) depicts results obtained for the west-east Stream $A \rightarrow B$ that is coordinated by the hierarchical approach during the first half of the simulation. Again, travel times and stops could be improved remarkably during that period. After the change of the traffic demand, the stream is less heavily used and therefore no longer coordinated. For the whole simulated period a reduction of 12.6% with respect to travel time and a 39.5% reduction of stops is obtained for this stream by hierarchical coordination. All reductions obtained with respect to travel times and stops for the network and for the investigated streams are summarised in Table II.

### B. Simulation Results

Figure 5 depicts the network-wide travel times and stops obtained from the simulations. It can be observed that both coordination approaches are able to reduce the number of stops in the network for (nearly) the complete simulated duration compared to an uncoordinated operation. After an initial learning phase that is needed to populate the rule sets of the intersections’ LCSs, an average reduction of 8.2% and 18.8% is obtained for the decentralised and the hierarchical approach, respectively. As expected for the investigated test case, the hierarchical approach achieves a higher reduction of stops, but the DPSS mechanism still performs reasonably well. The abrupt change in the traffic demand (indicated by the dotted vertical line) results in temporarily increased travel times and stops for both approaches until the change is detected and the coordination is adapted.

With respect to travel times no significant reductions were obtained by coordination (see Table II for details). This is due to the fact that coordination leads to increased cycle times for all but the most heavily used intersections within a PSS (see Step D.2 for details on cycle time determination). The increased cycle time of the less utilised intersections results in increased local delays whenever the intersections cannot benefit from coordination (i.e., when an intersection is the start of a PSS). These local delays contribute to the network-wide travel time.

To illustrate the differences between decentralised and hierarchical coordination, two selected streams have been evaluated with respect to travel times and stops. Figure 6(a) depicts the results obtained for Stream $G \rightarrow H$ which is the most heavily used stream during the first half of the simulation. During this period, the DPSS approach creates a PSS for this stream that leads to decreased travel times and stops. After the traffic demand changes at the beginning of the second half of the simulated period, the stream is no longer coordinated and as a result travel times and stops rise to a level similar to the one obtained for uncoordinated operation. Nevertheless, a reduction of 11.7% with respect to travel time and 22.0% with respect to stops is obtained for this traffic stream by decentralised coordination when the whole simulated period is considered.

Figure 6(b) depicts results obtained for the west-east Stream $A \rightarrow B$ that is coordinated by the hierarchical approach during the first half of the simulation. Again, travel times and stops could be improved remarkably during that period. After the change of the traffic demand, the stream is less heavily used and therefore no longer coordinated. For the whole simulated period a reduction of 12.6% with respect to travel time and a 39.5% reduction of stops is obtained for this stream by hierarchical coordination. All reductions obtained with respect to travel times and stops for the network and for the investigated streams are summarised in Table II.

### Table II

<table>
<thead>
<tr>
<th>Stream</th>
<th>Travel time</th>
<th>Stops</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G \rightarrow H$</td>
<td>11.7%</td>
<td>22.0%</td>
</tr>
</tbody>
</table>

Additionally, the environmental impact of the proposed coordination mechanisms is an important issue. AIMSUN’s environmental models have been used to evaluate the vehicles’ fuel consumption and their emission of pollutants. The models consider idling, accelerating, and decelerating periods of the simulated vehicles as well as different cruising speeds. Their configuration is according to AIMSUN’s manual using data from 1992 and 1994, respectively.

Figure 7 depicts the fuel consumption of the simulated vehicles over the simulation period. The figure shows that after a learning phase of approximately 20 min that is due to the initially empty rule set of the LCS, the fuel consumption could be reduced for both coordination approaches. Similar to the results obtained for travel times and stops, the worst fuel consumption of 14.5 l/100 km on average is obtained for the uncoordinated organic intersections. By decentralised coordination, this value could be reduced by 4.6% to 13.8 l/100 km, showing that the reduction of stops by coordination has a beneficial effect also on the consumed fuel. The best result is obtained for the hierarchical approach that leads to an average fuel consumption of 13.0 l/100 km which corresponds to a reduction of approximately 10.3% compared to the uncoordinated approach. The reductions

### Table II

<table>
<thead>
<tr>
<th>Stream</th>
<th>Travel time</th>
<th>Stops</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A \rightarrow B$</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>$G \rightarrow H$</td>
<td>11.7%</td>
<td>22.0%</td>
</tr>
</tbody>
</table>

Additionally, the environmental impact of the proposed coordination mechanisms is an important issue. AIMSUN’s environmental models have been used to evaluate the vehicles’ fuel consumption and their emission of pollutants. The models consider idling, accelerating, and decelerating periods of the simulated vehicles as well as different cruising speeds. Their configuration is according to AIMSUN’s manual using data from 1992 and 1994, respectively.

Figure 7 depicts the fuel consumption of the simulated vehicles over the simulation period. The figure shows that after a learning phase of approximately 20 min that is due to the initially empty rule set of the LCS, the fuel consumption could be reduced for both coordination approaches. Similar to the results obtained for travel times and stops, the worst fuel consumption of 14.5 l/100 km on average is obtained for the uncoordinated organic intersections. By decentralised coordination, this value could be reduced by 4.6% to 13.8 l/100 km, showing that the reduction of stops by coordination has a beneficial effect also on the consumed fuel. The best result is obtained for the hierarchical approach that leads to an average fuel consumption of 13.0 l/100 km which corresponds to a reduction of approximately 10.3% compared to the uncoordinated approach. The reductions
Fig. 5. Network-wide travel time and number of stops

Fig. 6. Travel time and number of stops for selected streams

Fig. 7. Fuel consumption for the simulation period
reported for fuel consumption directly map to the emission of Carbon Dioxide (CO₂), since for a given type of fuel a vehicle’s CO₂ emissions are directly proportional to the quantity of fuel consumed [16].

Regarding other pollutants, Carbon Monoxide (CO), Nitrogen Oxides (NOₓ), and un-burnt Hydrocarbons (HC) have been considered in the simulation study since they are the main pollutants emitted from petrol and diesel engines. Unlike CO₂, emissions of these pollutants are not directly linked to fuel consumption. CO, NOₓ, and HC are emitted especially during high load and idling periods of the engine, i.e., when vehicles are standing with running engines or when they have to accelerate after a stop.

Table III states the reduction of pollution emission obtained by the coordination of traffic signals. While the decentralised and hierarchically coordinated approaches are both able to reduce the emission of pollutants compared to an uncoordinated signal system, the hierarchical coordination obtains better results with respect to all pollutants. This improvement was to be expected since a higher reduction of stops could be obtained by hierarchical coordination which results in a reduced amount of high load phases for the acceleration of stopped vehicles.

TABLE III
REDUCTION OF EMISSIONS COMPARED TO UNCOORDINATED OPERATION

<table>
<thead>
<tr>
<th></th>
<th>CO</th>
<th>NOₓ</th>
<th>HC</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPSS RM</td>
<td>3.7%</td>
<td>7.0%</td>
<td>5.0%</td>
</tr>
<tr>
<td></td>
<td>5.2%</td>
<td>11.5%</td>
<td>3.3%</td>
</tr>
</tbody>
</table>

In summary, both presented traffic-adaptive coordination mechanisms have a beneficial impact on the traffic network. Especially the number of stops, the fuel consumption, and pollution emissions can be reduced by coordination. For the investigated test case – that was designed to exploit a weakness of the DPSS mechanism – the hierarchical coordination showed a better performance.

VI. CONCLUSION

The paper revisited a decentralised coordination mechanism for traffic signals in urban road networks. The mechanism supports the traffic-responsive creation of progressive signal systems and relies on local traffic demands in its decision process. An exemplary worst case scenario that exploits the mechanism’s locally limited view has been identified and investigated. To handle such special cases and to investigate the possibilities and limitations of decentralised traffic control systems, a novel hierarchical extension has been proposed. An additional component – the Regional Manager – supports the coordination of signalised intersections and handles special cases better than the purely decentralised solution. The decentralised and hierarchical approaches have been compared in a simulation study using the microscopic traffic simulator AIMSUN from the field of traffic engineering. The study focussed on traffic demands that are difficult to handle for the decentralised approach and presented benefits obtainable by the Regional Manager.

As future work, more complex coordination schemes that allow for bi-directional coordination or the handling of crossing progressive signal systems will be investigated. Furthermore, a comparison with existing traffic control systems based on a simulated real-world network is intended. A comparison to adaptive network control systems is desirable, but poses difficulties related to their simulation. Therefore, a next step will focus on a comparison with day-time dependent coordinations that are frequently used in Germany.

ACKNOWLEDGMENT

The authors would like to thank Fabian Rochner for his help. This work was supported by the German Research Foundation (DFG) within the priority programme 1183 “Organic Computing”.

REFERENCES