



HAL
open science

Capacity-aware back-pressure traffic signal control

Jean Gregoire, Emilio Frazzoli, Arnaud de La Fortelle, Tichakorn
Wongpiromsarn

► **To cite this version:**

Jean Gregoire, Emilio Frazzoli, Arnaud de La Fortelle, Tichakorn Wongpiromsarn. Capacity-aware back-pressure traffic signal control. 2013. hal-00865966v2

HAL Id: hal-00865966

<https://hal.science/hal-00865966v2>

Preprint submitted on 29 Oct 2013 (v2), last revised 5 May 2014 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Capacity-aware back-pressure traffic signal control

Jean Gregoire Emilio Frazzoli Arnaud de La Fortelle Tichakorn Wongpiromsarn

Abstract—The control of a network of signalized intersections is considered. A queuing network with exogenous arrivals is used for modelling. Previous works demonstrate that the so-called back-pressure control provides stability guarantees under infinite queues capacities. In this paper, we highlight the failing of state-of-the-art back-pressure control under finite capacities by identifying sources of non work-conservation and congestion propagation. We propose the use of a normalized pressure which guarantees work-conservation and mitigates congestion propagation while ensuring fairness at low traffic density and recovering original back-pressure as capacities grow to infinity. This capacity-aware back-pressure control enables to handle higher arrival rates, as indicated by simulation results, and keeps the key benefits of back-pressure: ability to be distributed over intersections and $\mathcal{O}(1)$ complexity.

I. INTRODUCTION

Congestion is one of the major problems in today’s metropolitan transportation networks. Before investigating investments in order to enhance the capacity of the network or policies to reduce the traffic, one must wonder whether the network is used at its maximum capacity. Cars automation is expected to enable much more precise and intelligent coordination between vehicles, reducing congestion [1]. However, automated cars are not currently ready for large commercial distribution. Human-driven cars can only be coordinated by traffic signals: more complex scheduling at intersections would require automation to be safe. That is why it is of high interest to study the theoretical maximum throughput of a network of intersections coordinated by traffic lights.

Traffic lights at intersections alternate the right-of-way of users (e.g., cars, public transport, pedestrians) to coordinate competing flows. A particular set of feasible simultaneous rights of way, called a phase, is decided for a period of time called the time slot [2]. Controlling a traffic light consists of designing rules to decide which phase to apply over time. It is convenient to use a fixed pre-defined time slot length, whose size corresponds to the minimal duration of a phase. Indeed, due to the dynamic constraints of the vehicles and to ensure safety, a phase cannot last less than a few seconds. When the time slot size is pre-defined, controlling the network only consists of deciding which phase to apply at every time slot.

Pre-timed policies activate phases according to a time-periodic pre-defined schedule. There is much previous work on designing optimal pre-timed policies. However, such policies are not efficient under changing arrival rates which require adaptive control. Most used adaptive traffic signal control systems include SCOOT [3], SCATS [4], PROLYN [5], RHODES [6], OPAC [7] or TUC [8]. These systems update

some control variables of a configurable pre-timed policy on middle term, based on traffic measures, and apply it on short term. Control variables may include phases, splits, cycle times and offsets [2]. Such algorithms may differ in the way optimization is carried out (e.g. linear/dynamic programming, exhaustive enumeration) and the modeling (e.g., queuing network model [9], cell transmission model [10], store-and-forward [11], petri nets [12]). Many major cities currently employ these systems which proved to be able to produce various benefits, including travel time and fuel consumption reduction, as well as safety improvements [13].

More recently, based on the seminal paper [14], feedback controls have been proposed both in the case of deterministic arrivals [15], or stochastic arrivals [16], [17]. Control is not limited to the control variables mentioned above and the phase to apply at every time slot is computed based on queue length estimation. This requires real-time queues estimation, but it enables to be much more reactive than other traffic controllers and to have stability guarantees. [14] has introduced the so-called back-pressure control which computes the control to apply based on queues lengths, and can achieve provably maximum stability. This algorithm was originally applied to wireless communication networks [18], [19], and some effort has been required to apply the approach in the context of a network of intersections [16], [17]. A key feature of this algorithm is that it can be completely distributed over intersections, it requires only local information and it is of $\mathcal{O}(1)$ complexity.

However, the strong assumption of state-of-the-art back-pressure traffic control is the infiniteness of queues capacities. Indeed, when the queue at the entry of an intersection grows so much that it reaches the upstream intersection, congestion will propagate: this is a non-negligible and easy to observe phenomenon. The phenomenon is commonly referred as blocking in queuing theory, and many blocking types can be considered [20]. In worst-case scenario, blocking results in deadlocks whose resolution can be of high complexity [21], [22]. The pre-timed policy proposed in [9] takes into account queue capacities in the optimization explicitly. Some works applied to wireless communication networks have proposed feedback controls that can achieve maximum throughput under queue boundedness constraints. However, they suppose the absence of arrivals at internal nodes, cannot be easily implemented [23] and are thus not suitable for our application.

This paper proposes to keep the the fundamental idea of back-pressure control, that is pressure computation at every node of the network, in order to keep the resulting key benefits: ability to be distributed over intersections and $\mathcal{O}(1)$

complexity. After identifying the sources of inefficiency of state-of-the-art back-pressure control, we introduce capacity-aware pressure functions to mitigate congestion propagation.

The paper is organized as follows. Section II describes the queuing network model. Section III presents the state-of-the-art back-pressure traffic signal control of [17] and proves its non work-conservation and its inability to avoid congestion propagation. Section IV proposes the use of normalized pressures and proves the benefits in terms of work-conservation and congestion mitigation. Simulations of Section V show the efficiency of the approach proposed in this paper and Section VI concludes and opens perspectives.

II. MODEL

A. Queuing network topology

The network of intersections is modelled as a directed graph of nodes $(N_a)_{a \in \mathcal{N}}$ and links $(L_j)_{j \in \mathcal{L}}$. The graph is referred as the network graph. Nodes represent lanes with queuing vehicles, and links enable transfers from node to node. This is a standard queuing network model.

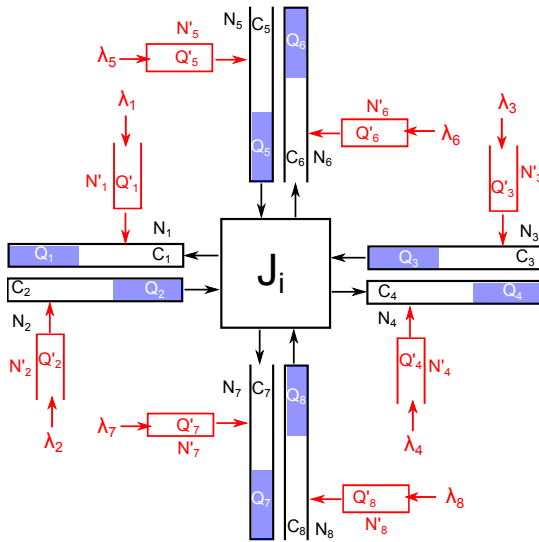


Fig. 1. A junction with 4 incoming nodes and 4 outgoing nodes which corresponds to the intersection depicted in Figure 2.

A key point for our application in traffic signal control is that it is a multiple queues one server queuing network. Every signalized intersection is modelled as a server managing a junction which consists of set of links. Junctions $(J_i)_{i \in \mathcal{J}}$ are supposed to form a partition of links. For every junction J , $\mathcal{I}(J)$ and $\mathcal{O}(J)$ denote respectively the inputs and the outputs of J . Inputs (resp. outputs) of junction J are nodes N such that there exists a link $L \in J$ pointing from (resp. to) N . The reader should consider the introduction of junctions in the model as an overlay of the queuing network model.

For the sake of simplicity, we do not represent links in the queuing network representation of Figure 1 and we assume that at every junction, there exists a link from any input to any output. If the link does not exist physically, the flow through this link will be constrained to zero.

Every server maintains an internal queue for every input/output, and server work enables to transfer vehicles from an input to an output of the junction. Due to routing of vehicles, the internal queue at node N_a is a vector Q_a and $Q_{ab}(t)$ denotes the number of vehicles in the queue of node N_a entering N_b upon leaving N_a . Time is slotted: $t \in \mathbb{N}$, and the aggregated queue length $Q_a(t) = \sum_b Q_{ab}(t)$ denotes the total number of vehicles at node N_a considering all possible routings after exiting N_a .

In this paper, queues are supposed to have finite capacities.

Definition 1 (Queue capacity, Congested node). *Every node N_a is supposed to have an intrinsic queue capacity, denoted C_a . We say C_a is the capacity of N_a .*

If $Q_a > Q_a^{\text{lim}} = C_a - \Delta Q_a^{\text{max}}$, we say N_a is congested, where ΔQ_a^{max} denotes the maximum endogenous input transfer to N_a in a time slot. We assume $C_a \geq \Delta Q_a^{\text{max}}$, and Q_a^{lim} is referred as the congestion threshold.

The definition of a congested node ensures that when a node is not congested, it can always accept vehicles entries, because the margin ΔQ_a^{max} equals the maximum endogenous input transfer. Indeed, even if the output flow is zero, Q_a will not grow over C_a .

B. Phase-based control

At every time slot t , servers work, resulting in vehicles transfers. It is convenient to consider the service matrix μ defined below:

Definition 2 (Service rate, Flow). *For all $a, b \in \mathcal{N}$,*

- *the service rate μ_{ab} represents the transmission rate offered by servers to transfer vehicles from N_a to N_b , i.e. the maximum number of vehicles transferred from N_a to N_b during the next time slot;*
- *the endogenous flow variable f_{ab} represents the actual number of vehicles leaving N_a and entering N_b .*

Since the number of vehicles transferred is less or equal to the transmission rate offered by the servers, the following inequality holds:

$$f_{ab} \leq \mu_{ab} \quad (1)$$

Assumption 1 (Feasible vehicles transfers). *Only the vehicles which are currently at a node at the beginning of time slot t can be transferred from that node to another node during slot t .*

Let define the input/output rate with regards to a given node N_a associated to a matrix g .

Definition 3 (Input rate, Output rate). *Given a matrix g , for all $a \in \mathcal{N}$, the input rate g_a^{in} and the output rate g_a^{out} with regards to N_a are defined as follows:*

$$g_a^{\text{in}} = \sum_b g_{ba} \quad (2)$$

$$g_a^{\text{out}} = \sum_c g_{ac} \quad (3)$$

If μ is the service matrix, and f the flow matrix, the following inequalities hold:

$$f_a^{\text{in}} \leq \mu_a^{\text{in}} \quad (4)$$

$$f_a^{\text{out}} \leq \mu_a^{\text{out}} \quad (5)$$

Under phase-based control, service rates are set by activating a given signal phase p_i at every junction J_i from a predefined finite set of feasible phases \mathcal{P}_i at every time slot t . Each global phase $p = (p_i)_{i \in \mathcal{J}} \in \mathcal{P}$ results in a different service $\mu(p)$ where $\mathcal{P} = \prod_{i \in \mathcal{J}} \mathcal{P}_i$ denotes the set of feasible global phases.

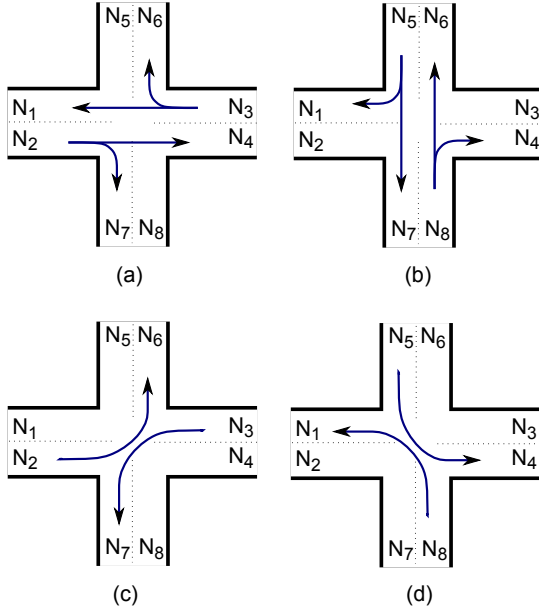


Fig. 2. A typical set of feasible phases at a junction. For example, supposing that service rates equal 0 or 1, the non zero service rates for phase (a) are μ_{31} , μ_{36} , μ_{24} and μ_{27}

Assumption 2 (Phase-controlled service). *If $N_a \in \mathcal{I}(J_i)$ and $N_b \in \mathcal{O}(J_i)$, the service rate μ_{ab} satisfies:*

$$\mu_{ab} \in \{\mu_{ab}(p_i) : p_i \in \mathcal{P}_i\} \quad (6)$$

The service matrix μ satisfies:

$$\mu \in \{\mu(p) : p \in \mathcal{P}\} \quad (7)$$

The abuse of notation in the above assumption is justified by the fact that the service rate depends only on the applied phase p , i.e. can also be considered as a function of p .

Figure 2 depicts the 4 typical phases of a 4 inputs/4 outputs junction. We define linked nodes as nodes such that there exists a feasible phase transferring vehicles from one node to the other one:

Definition 4 (Linked nodes). *N_a and N_b are linked at junction J_i if there exists a phase $p_i \in \mathcal{P}_i$ such that $\mu_{ab}(p_i) > 0$.*

In this paper, for the sake of simplicity, we do not take into account any exogenous variable z which would affect the flow matrix associated to each phase, yielding a service matrix $\mu(z, p)$. However, as proved in previous works [17],

[19], back-pressure properties can be extended to this case. As a result, we assume that for each phase the service rate from one node to another node is zero or equals the saturation rate:

Assumption 3 (Binary service rates). *For all $a, b \in \mathcal{N}$, there exists s_{ab} , the saturation rate from N_a to N_b , such that for all $p \in \mathcal{P}$, $\mu_{ab}(p) \in \{0, s_{ab}\}$.*

Finally, we make the following assumption which enables to switch off the service from N_a to N_b independently from other services:

Assumption 4 (Service rates independence). *For all phase $p \in \mathcal{P}$ and for all $a, b \in \mathcal{N}$, there exists a phase $\tilde{p} \in \mathcal{P}$ such that $\mu_{ab}(\tilde{p}) = 0$ and for all $(c, d) \neq (a, b)$, $\mu_{cd}(\tilde{p}) = \mu_{cd}(p)$.*

C. Exogenous arrivals

We assume that there is no exogenous arrival at nodes N_a for all $a \in \mathcal{N}$. However, every node N_a is associated to a node N'_a of infinite capacity where exogenous arrivals occur, and there is a link from N'_a to N_a transferring exogenous arrivals at N'_a into N_a . Q'_a denotes the length of the queue at N'_a and the dynamics of the link from N'_a to N_a is described below.

N'_a maintains a queue, and when an exogenous arrival occurs during slot t ,

- if N_a is not congested at the end of slot t , the vehicle is directly transferred into N_a ;
- otherwise, the vehicle stays in N'_a , until N_a gets non-congested at the end of time slot $t' > t$.

Let $A_a(t)$ denote the number of vehicles that exogenously arrive at node N'_a during slot t .

Definition 5 (Rate convergent process). *A process $X(t)$ is rate convergent with rate x if:*

- $\lim_{T \rightarrow +\infty} \frac{1}{T} \sum_{t=0}^{T-1} X(t) = x$
- For any $\delta > 0$, there exists an interval size T such that for any initial time t_0 and regardless of pas history, the following condition holds: $|\mathbb{E}\{\frac{1}{T} \sum_{t=0}^{T-1} X(t)\} - x| \leq \delta$

Assumption 5 (Rate convergent arrival process). *For all $a \in \mathcal{N}$, the process $A_a(t)$ is rate convergent with rate $\lambda_a \geq 0$. For all t , $A_a(t)$ is independent from $\{Q(\tau)\}_{\tau \leq t}$.*

Finally, we assume that controllers do not have access to buffer queues lengths Q'_a and can only rely on the internal queue matrix Q to compute the control to apply. This is a realistic assumption in the absence of vehicle-to-infrastructure communications.

D. Blocking

We assume that only balanced flows with regards to a node are admissible as soon as this node is congested, as stated in the definition below. Basically, if a node is congested, the incoming flow must be lower than the outgoing flow.

Definition 6 (Admissible flow). *The flow f is admissible if for all congested nodes N_a ,*

$$f_a^{\text{in}} \leq f_a^{\text{out}} \quad (8)$$

Algorithm 1 transforms any flow matrix f into a feasible flow matrix while minimizing flow variables removals as stated by Property 2. The principle of flow reduction is depicted in Figure 3.

Algorithm 1 Optimal flow reduction

Require: Flow vector f

```

function OPTIMALFLOWREDUCTION
  do
    recall  $\leftarrow$  false
    for  $a$  s.t.  $N_a$  is congested do
      5: while  $f_a^{\text{in}} > f_a^{\text{out}}$  do
        Choose any  $b$  such that  $f_{ba} > 0$ 
         $f_{ba} \leftarrow f_{ba} - \min(f_{ba}, f_a^{\text{in}} - f_a^{\text{out}})$ 
        recall  $\leftarrow$  true
      end while
    10: end for
    while recall
      return  $f$ 
    end function
  
```

Property 1 (Flow reduction algorithm termination and correctness). *Algorithm 1 terminates and the returned flow is admissible.*

Proof. As long as Algorithm 1 is running, at every iteration of the do loop there exists a congested node N_a such that $f_a^{\text{in}} > f_a^{\text{out}}$. Since flow variables are integers, at every iteration, at least one flow variable f_{ba} is decreased of one unit at Line 7. As a result, after at most $\sum_{ab} f_{ab}$ iterations, Algorithm 1 necessarily terminates, because f would be zero and for all nodes N_a , $0 = f_a^{\text{in}} \leq f_a^{\text{out}} = 0$. Moreover, if Algorithm 1 terminates, then for all congested nodes N_a , $f_a^{\text{in}} \leq f_a^{\text{out}}$, so the returned flow is admissible. \square

Property 2 (Optimal flow reduction). *For any flow f , Algorithm 1 returns an optimally reduced admissible flow f^* associated to f :*

Assume a flow matrix \tilde{f} satisfies:

$$\tilde{f} \neq f^* \quad (9)$$

$$\forall a, b \in \mathcal{N}, \tilde{f}_{ab} \leq f_{ab} \quad (10)$$

$$\forall a, b \in \mathcal{N}, \tilde{f}_{ab} \geq f_{ab}^* \quad (11)$$

Then \tilde{f} is not admissible.

Proof. If $f^* = f$, the returned flow is obviously optimally reduced. Suppose $f^* \neq f$ and there exists an admissible flow \tilde{f} satisfying Equations 9, 10 and 11. Say $\tilde{f}_{ab} > f_{ab}^*$. Equation 9 implies that $\tilde{f}_{ab} \neq f_{ab}$. Then, N_b is necessarily congested, the flow variable from N_a to N_b has necessarily been reduced at Line 7 of Algorithm 1, and by construction, we have $f_b^{\text{out}} = f_b^{\text{in}}$. As a consequence, if Equations 10 and 11 are satisfied, we necessarily have $\tilde{f}_{ab}^{\text{in}} > \tilde{f}_{ab}^{\text{out}}$ and \tilde{f} is not admissible. Note that there is not a unique optimally reduced flow, but several, and Algorithm 1 returns one of them. \square

In reality, if the traffic signal gives the right-of-way to vehicles going to a congested node, and if the node is still congested at the end of the time slot, some vehicles could

stay stuck in the intersection area with huge consequences on the flow in the next time slot. This can be partly prevented by rules of the road. That is why Algorithm 1 should also be considered as an operation which should be carried out on the service matrix at the beginning of every time slot in order to remove the right-of-way to vehicles which risk to get stuck in the intersection at the end of the time slot.

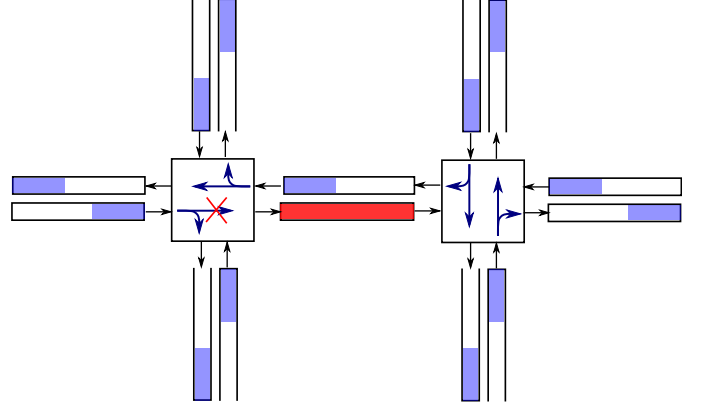


Fig. 3. Optimal flow reduction by Algorithm 1. Since the red colored middle node is congested and not emptied by the phase at the right junction, the flow filling this node at the left junction is removed.

In the remainder of the paper, we assume that the service $\mu(p(t))$ associated to a control $p(t)$ can be non-admissible, but the flow $f(t)$ applied at junctions will be reduced by Algorithm 1 to be admissible. Note that Algorithm 1 does not specify a deterministic way to choose the order to browse congested nodes at Line 4, nor the way to choose b at Line 6. In the remainder of the paper, we assume that the implementation of the algorithm is deterministic and the flow returned by Algorithm 1 is fixed for a given input.

Note that the above model gives priority to endogenous flow since exogenously arriving vehicles are accepted to "really enter" the network if and only if the node is non-congested at the end of the time slot, i.e. after endogenous arrivals.

E. Routing model and network dynamics

When a quantity of vehicles arrives at node $N_a \in \mathcal{I}(J_i)$ during slot t , exogenously and endogenously, it is split and added into queues Q_{ab} , $b \in \mathcal{O}(J_i)$, according to an exogenous routing process $R_{ab}^{(t)}$, defined for all $a, b \in \mathcal{N}$.

Assumption 6 (Rate convergent routing process). *The arrival process and the routing process are independent, and for all t , $R_{ab}^{(t)}$ is independent from $\{Q(\tau)\}_{\tau \leq t}$. $R_{ab}^{(t)}$ takes an integer, returns an integer, and for $X \in \mathbb{N}$, $\sum_b R_{ab}^{(t)}(X) \leq X$.*

For all process $X(t)$ such that for all t , $R_{ab}^{(t)}$ is independent from $\{X(\tau)\}_{\tau \leq t}$, there exists a rate $r_{ab} \geq 0$ for all $a, b \in \mathcal{N}$ such that $R_{ab}^{(t)}(X(t)) - r_{ab}X(t)$ is rate convergent with rate 0. As a consequence of the above assumptions:

$$\sum_b r_{ab} \leq 1 \quad (12)$$

Note that $1 - \sum_b r_{ab}$ represents the exit rate of vehicles entering node N_a . One could consider an additional node w

representing the external world playing the role of sink of the exit flow from N_a at rate $r_{a\omega} = 1 - \sum_b r_{ab}$

The latter routing assumption closes our model, and the dynamics of the network is now fully described. Since service rates depend only on the phase applied at every junction, controlling the network consists of controlling the phase applied at every junction.

Definition 7 (Control). A control $p(t)$ for the queuing network returns the phase to apply at every junction during slot t . If $p(t)$ depends only on the state $Q(t)$ of the network at time slot t , $p(t)$ is a feedback control.

The network dynamics under control p follows:

$$Q'_a(t+1) = \max(0, Q_a(t) + \mu_a^{\text{in}}(p(t)) - \mu_a^{\text{out}}(p(t)) + Q'_a(t) + A_a(t) - C_a) \quad (13)$$

$$Q_{ab}(t+1) = Q_{ab}(t) + R_{ab}^{(t)}(Q'_a(t) + A_a(t) - Q'_a(t+1) + f_a^{\text{in}}(t)) - f_{ab}(t) \quad (14)$$

The flow $f(t)$ is the optimally reduced flow returned by Algorithm 1 taking as input flow $g(t)$ defined below:

$$\forall a, b \in \mathcal{N}, g_{ab}(t) = \min(Q_{ab}(t), \mu_{ab}(p(t))) \quad (15)$$

If $p(t)$ is a feedback control, i.e. a function of $Q(t)$, the process $(Q(t), Q'(t))$ is a Markov chain with long-term stationary transition probabilities, which depend on the feedback control.

III. FAILING OF BACK-PRESSURE CONTROL UNDER BOUNDED QUEUES CONSTRAINTS

A. Back-pressure control

[16] proposes a stability-optimal back-pressure control under infinite capacities. However, the proposed control requires complete knowledge of the queues lengths matrix $Q(t)$ and knowledge of the routing rates. In contrast, the back-pressure control proposed in [17] uses only the aggregated queues lengths $Q_a(t) = \sum_b Q_{ab}(t)$. For our application, a complete knowledge of the queues lengths matrix $Q(t)$ is not realistic because dedicated lanes for turning vehicles are only at the proximity of the junction. Farther, all vehicles are gathered and the controller does not have access to the direction of every vehicle in the absence of vehicle-to-infrastructure communications.

The back-pressure control with unknown routing rates of [17] is defined by Algorithm 2. It computes the phase to apply at every time slot without requiring neither routing rates nor complete knowledge of queues lengths matrix $Q(t)$ and takes as inputs the aggregated queues lengths $Q_a(t) = \sum_b Q_{ab}(t)$. However, it still requires vehicle detectors variables $d_{ab}(t) \in [0, 1]$ defined below:

$$d_{ab}(t) = \frac{\max(s_{ab}, Q_{ab}(t))}{s_{ab}} \quad (16)$$

The variable $d_{ab}(t)$ is easier to measure than $Q_{ab}(t)$ because it only requires the knowledge of $Q_{ab}(t)$ in the range $[0, s_{ab}]$, i.e. at proximity of the junction. Note that given the binary service rate assumptions, the following equality holds in the absence of blocking:

$$f_{ab}(t) = d_{ab}(t)\mu_{ab}(p(t)) \quad (17)$$

Algorithm 2 Back-pressure control

Require:

Aggregated queues lengths $Q_a(t)$ for all $a \in \mathcal{N}$,
 Pressure functions $P_a(Q_a)$ for all $a \in \mathcal{N}$,
 vehicle detectors variables $d_{ab}(t)$,
 Phase selection policy ϕ in case of equality.

5: **function** BACKPRESSURECONTROL

for $i \in \mathcal{J}$ **do**

for $a \in \mathcal{I}(J_i) \cup \mathcal{O}(J_i)$ **do**

$\Pi_a(t) = P_a[Q_a(t)]$

end for

10: **for** $a \in \mathcal{I}(J_i), b \in \mathcal{O}(J_i)$ **do**

$W_{ab}(t) = d_{ab}(t) \max(\Pi_a(t) - \Pi_b(t), 0)$

end for

$p_i^*(t) \leftarrow \arg_{\phi} \max_{p_i \in \mathcal{P}_i} \sum_{a \in \mathcal{I}(J_i), b \in \mathcal{O}(J_i)} W_{ab}(t) \mu_{ab}(p_i)$

end for

15: **return** Phase $p^*(t)$ to apply in time slot t

end function

Algorithm 2 is a generic version of back-pressure. Indeed, it does not specify neither pressure functions, nor the policy ϕ that decides which phase to select at the $\arg \max$ of Line 13 when an equality holds. For example, the policy ϕ^{random} consists of selecting randomly any phase which maximizes the weighted sum.

The two below properties expose two key benefits of back-pressure control: ability to be distributed over junctions and $\mathcal{O}(1)$ complexity.

Property 3 (Distribution over junctions). Back-pressure control computation can be distributed at every junction, requiring only inputs/outputs queues lengths.

Property 4 ($\mathcal{O}(1)$ complexity). Algorithm 2 computes back-pressure control in $\mathcal{O}(1)$ complexity.

State-of-the-art back-pressure traffic signal control [17] uses Algorithm 2 with linear pressure functions $P_a(Q_a) = Q_a$. It is proved in [16] and [17] that stability guarantees of back-pressure control can be obtained under infinite queues capacities.

However, under bounded queues constraints, pressure at N_a saturates at $P_a = C_a$ for $Q_a = C_a$. In the following, we show that this saturation at different levels for every node may result in non work-conservation and congestion propagation as presented in this section.

B. Non work-conservation

First of all, let define the notion of work and work-conservation in our context.

Definition 8 (Work). We say that the server at junction J_i works during slot t if flow variables at the junction are not all zeros during the slot.

Definition 9 (Work-conserving control). A control is work-conserving if a sufficient condition for the server of a junction to work during slot t is the existence of an input N_a and an output N_b such that $Q_{ab}(t) > 0$ and $Q_b(t) < Q_b^{\text{lim}}$.

Note that this definition does not mean that every server will work if there exists one control under which it works. Indeed, if all outputs are congested, it is still possible for the server to work, at the condition that other servers empty one of the congested outputs in the same time.

Non work-conservation is a sign of inefficiency for a given control. Due to limited queues capacities, back-pressure control under linear pressure functions is not work-conserving as stated in Theorem 1, with a concrete example depicted in Figure 4.

Theorem 1 (Non work-conservation under back-pressure). Under bounded queues constraints, the back-pressure control with unknown routing rates as defined by Algorithm 2 is not work-conserving in the general case.

Proof. Consider the network of Figure 4. Suppose that the middle junction has two feasible phases: p_{ab} with a unique non zero service rate from N_a to N_b , and p_{cd} with a unique non zero service rate from N_c to N_d (assume similarly that the right junction has two feasible phases p_{bg} and p_{ef}). Suppose that N_b is congested, N_d is non-congested, $Q_a > C_b$, $Q_c < Q_d$, $Q_e > Q_f$ and $C_b < Q_g$. Then, $\sum_{a'b'} W_{a'b'} \mu_{a'b'}(p_{ab}) = W_{ab} \mu_{ab}(p_{ab})$ and $\sum_{a'b'} W_{a'b'} \mu_{a'b'}(p_{cd}) = W_{cd} \mu_{cd}(p_{cd}) = 0$. So, the phase to apply at the middle junction computed by Algorithm 2 is p_{ab} . Similarly, the phase to apply at the right junction computed by Algorithm 2 is p_{ef} . Since, N_b is congested and not emptied by the right junction, the flow reduction to an admissible flow will remove the flow variable from N_a to N_b . As a result, the middle junction will not work. However, choosing phase p_{cd} would have enabled the server to work. \square

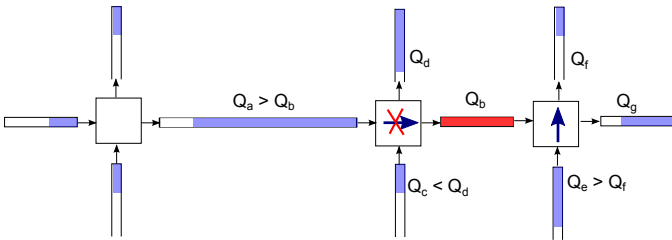


Fig. 4. Example of non work-conservation. N_b is congested so its occupancy Q_b is bounded to C_b . N_a is not congested and has a higher capacity than N_b , so its occupancy Q_a is greater than Q_b . Given current queues lengths, the phase p computed by 2 using linear pressures is such that $\mu_b^{\text{out}}(p) = 0$ and $\mu_{ab}(p) > 0$. N_b being congested, the reduction to an admissible flow will remove flow $\mu_{ab}(p)$ and the middle node of the drawing will not work.

The proved non work-conservation is an important property since the subsequent inefficiency results in congestion propagation as highlighted in the following.

C. Congestion propagation and deadlocks

As depicted in Figure 5, if the phase returned by the control is not admissible, it will result in congestion propagation, both to the node which has the right-of-way but cannot empty because of downstream congestion, and to the node which has not the right-of-way.

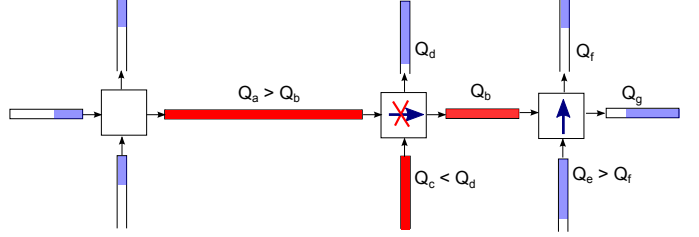


Fig. 5. Congestion propagation due to non work-conservation. Since N_b is congested and not emptied, flow $\mu_{ab}(p)$ is removed, N_a is not emptied, so congestion propagates to N_a . Moreover, N_c is also not emptied, so congestion propagates to both nodes N_a and N_c .

In worst-case scenario, such congestion propagation can lead to deadlocks, as depicted in Figure 6.

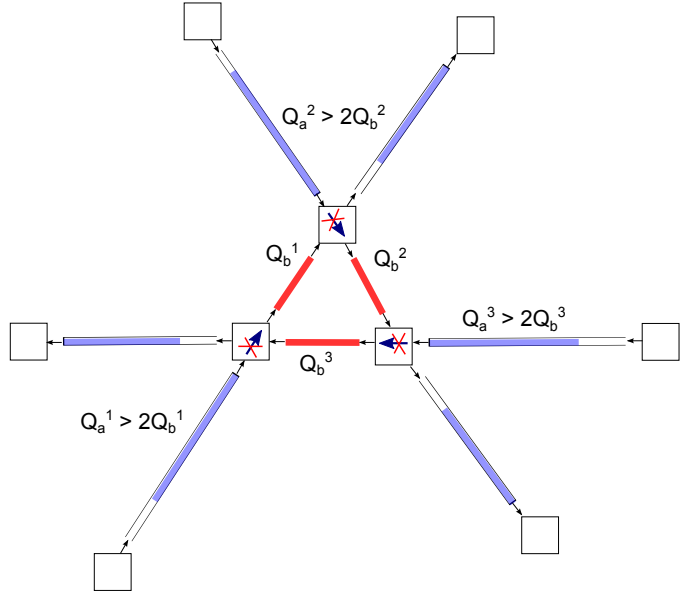


Fig. 6. A deadlock for back-pressure control. All applied phases are inadmissible and Q_a^1 , Q_a^2 and Q_a^3 will be ever growing.

One must distinguish between weak deadlocks as the one depicted in Figure 6 that can be easily locally resolved by applying an appropriate phase control, and strong deadlocks that cannot be resolved at all due to routing decisions of vehicles. The two sorts of deadlocks are formally defined below:

Definition 10 (Weak/Strong deadlock). A state Q^0 is a strong deadlock if for all possible controls p , the Markov chain with initial state Q^0 under null arrival process $A(t) = 0$ does not reach zero in finite time, i.e.:

$$\forall t \geq 0, (Q, Q')(t) \neq 0 \quad (18)$$

A state Q^0 is a weak deadlock for the control \tilde{p} if the Markov

chain under control \tilde{p} with initial state Q^0 and null arrival process $A(t) = 0$ does not reach zero in finite time.

Note that a state can be a weak deadlock for a given control and not for another one. In this paper, we modify state-of-the-art back-pressure control to mitigate congestion propagation by ensuring work-conservation. Congestion mitigation is expected to inhibit deadlocks to occurs due to less blocking.

IV. CAPACITY-AWARE TRAFFIC CONTROL

The following presents our approach for taking into account the limited queues capacities by using normalized pressures in back-pressure control. First of all, let introduce the reasons that motivate devising convex normalized pressures.

A. Purpose of a convex normalized pressure and criteria

1) *Purpose of a convex pressure:* When a node approaches full occupancy, every additional vehicle is more and more problematic as the queue grows. That is why it makes sense to define a pressure such that the marginal pressure, i.e. the increase in pressure due to an additional vehicle, rises as the queue grows. This remarks justifies the use of a convex pressure.

2) *Purpose of a normalized pressure:* When a node N_b is congested, it does not make sense to have a node N_a such that $P_a - P_b > 0$, since the "tap" associated to the flow from N_a to N_b cannot be opened. That is why we propose to normalize the pressures so that any congested node will exert a pressure that equals 1, while an empty node does not exert any pressure.

The most simple normalization that could be carried out would consist of using relative pressure functions $P_a(Q_a) = Q_a/Q_a^{\text{lim}}$. However, in order to have convex pressures and to respect the fairness requirement proposed below, the normalization will be slightly more complex.

3) *Requirement for fairness at low traffic density:* At low traffic density, there is no reason to be unfair. Indeed, even a random choice of phases would stabilize the network. So, unfairness would not be justified by any global stabilization goal. Suppose that we use relative pressures functions $P_a(Q_a) = Q_a/Q_a^{\text{lim}}$, then an additional vehicle causes an increase in pressure of $1/Q_a^{\text{lim}}$, i.e. as high as capacity decreases. That is why we define fair pressure functions, so that at low traffic density the marginal pressure should be uniform over nodes.

Definition 11 (Fair pressure functions at low traffic density). *We say that pressure functions $\{P_a(Q_a) : a \in \mathcal{N}\}$ are fair at low traffic density if:*

$$\exists K > 0 : \forall a \in \mathcal{N}, \frac{dP_a}{dQ_a}(0) = K \quad (19)$$

4) *Requirements for stability guarantees conservation:* Finally, it is important to ensure that as capacities grow to infinity, the original back-pressure control is recovered, to take advantage of the stability guarantees in the context of infinite capacities. That is why a requirement for the pressure function $P_a(Q_a)$ is to be linear for $Q_a/C_a \rightarrow 0$. If pressure functions are fair, $P_a(Q_a) = KQ_a + o_a(Q_a/C_a)$ in Landau notation, and

the pressure function is linear at low occupancy. As a result, if the queues are always much under maximum occupancy, i.e. if the infinite queues capacities assumption is valid, the back-pressure control and its stability guarantees are recovered.

Now we have presented criteria that should respect the modified pressure in order to be capacity-aware and fair at low traffic density, we propose in the following a convex normalized pressure which respects the above criteria.

B. Example of normalized pressure

The proposed pressure function should just be considered as an example of a pressure function fulfilling the presented requirements:

$$P_a(Q_a) = \min \left(1, \frac{\frac{Q_a}{C_\infty} + \left(2 - \frac{Q_a}{C_\infty}\right) \left(\frac{Q_a}{Q_a^{\text{lim}}}\right)^m}{1 + \left(\frac{Q_a}{Q_a^{\text{lim}}}\right)^{m-1}} \right) \quad (20)$$

At low occupancy, the pressure at node N_a is linear: $P_a(Q_a) \simeq Q_a/C_\infty$, so pressure functions are fair and respect the requirement for stability guarantees conservation. The function is convex: the slope of the pressure increases as occupancy grows. Pressure over congestion threshold is normalized: $\forall a \in \mathcal{N}, \forall Q_a \geq Q_a^{\text{lim}}, P_a(Q_a) = 1$. The shape of pressure functions for two different capacities is depicted in Figure 7. One can observe that the pressure function leaves the initial linear behaviour at lower occupancy as capacity decreases.

m and C_∞ are two parameters of the pressure functions. m configures the shape of the transition from the linear regime. C_∞ configures the slope of the pressure at low occupancy and is such that a node which capacity is C_∞ will have a linear pressure. We assume that all capacities are lower than C_∞ and $m > 1$.

C. Effect on queues length distribution

Suppose we apply back-pressure to a network of infinite queues capacities using pressure functions of Equation 20, C_a being only considered as positive numbers associated to every node. Assume the network is stabilized. Then, compared to the behaviour under linear pressure functions $P_a(Q_a) = Q_a$, the effect of the normalized pressures is a modification of the distribution of queues lengths over nodes. In particular, time averaged queues lengths can be expected to be smaller than under linear pressure for low capacity nodes, and higher for high capacity nodes. This modification of queues lengths distribution is expected to decrease the probability of congestion at low capacity nodes, resulting in less blocking and thus higher flows.

Of course, we cannot claim that the proposed control stabilizes the system under finite capacities constraints. Such a stability guarantee is out of the scope of this paper. However, since the control is equivalent to state-of-the-art back-pressure control as capacities grow to infinity, stability properties under infinite capacities are conserved. The control proposed in this paper improves the behaviour of the queuing network under finite capacities, as work-conservation property and

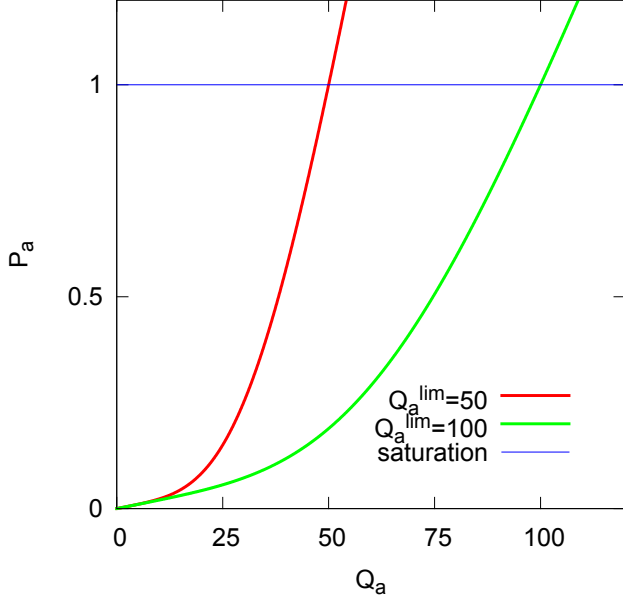


Fig. 7. Plot of the convex normalized pressure function with parameters $C_\infty = 500$ and $m = 4$ for two different congestion thresholds $Q_a^{\text{lim}} = 50$ and $Q_a^{\text{lim}} = 100$.

simulations in the following of the paper tend to indicate, while conserving stability properties under infinite capacities.

D. Work-conservation

As expected, pressure normalization enables to ensure work-conservation. This can be easily visualized in Figure 8 where one can observe that the weak deadlock of Figure 6 is resolved. In the following, we prove Theorem 2 which states work-conservation under convex normalized pressure.

Theorem 2 (Work-conservation under normalized pressures). *Assume that pressure functions are normalized, i.e. satisfy the below conditions for all $a \in \mathcal{N}$ (as the example of Equation 20):*

- P_a is an increasing function taking values in $[0, 1]$,
- $P_a(Q_a^{\text{lim}}) = 1$.

Assume also that the phase selection policy ϕ in case of equality privileges phases with a non zero service rates from N_a to N_b if $Q_{ab}(t) > 0$ and $Q_b(t) \leq Q_b^{\text{lim}}$.

Then, under finite capacities constraints, the back-pressure control defined by Algorithm 2 is work-conserving.

Proof. Suppose that back-pressure control using normalized pressure functions is not work-conserving for a given network at a given state. Then, there exists a server at a junction which does not work during some slot t , while it has an input N_a and an output N_b satisfying $Q_{ab}(t) > 0$ and $Q_b(t) \leq Q_b^{\text{lim}}$.

Let \tilde{p} denote the phase computed by Algorithm 2 using normalized pressure functions and let p denote a phase such that $\mu_{ab}(p) > 0$ (p exists due to service rates independence asserted in Assumption 4). If the server at the junction does not work after flow reduction by Algorithm 1, then for all non zero service rates $\mu_{a'b'}(\tilde{p}(t))$, there are two options:

- 1) $Q_b(t) > Q_b^{\text{lim}}$: the flow from N_a to N_b is removed by flow reduction of Algorithm 1,
- 2) or, $Q_{a'b'}(t) = 0$, there is no flow, because no vehicle in N_a enters N_b upon leaving N_a .

In the first case, $P_b(Q_b(t)) = 1$ and since P_a takes values in $[0, 1]$, we necessarily have $W_{a'b'}(t) = 0$. In the second case, $d_{a'b'}(t) = 0$, and we also have $W_{a'b'}(t) = 0$. As a result, $\sum_{a'b'} W_{a'b'} \mu_{a'b'}(\tilde{p}) = 0$.

On the other hand, by positivity of flow variables and weights, we have $\sum_{a'b'} W_{a'b'} \mu_{a'b'}(p) \geq 0$. If $\sum_{a'b'} W_{a'b'} \mu_{a'b'}(p) > 0$, it is absurd because p should have been output by Algorithm 2 instead of \tilde{p} . If $\sum_{a'b'} W_{a'b'} \mu_{a'b'}(p) = 0$, it is absurd because $\sum_{a'b'} W_{a'b'} \mu_{a'b'}(p) = \sum_{a'b'} W_{a'b'} \mu_{a'b'}(\tilde{p})$ and p should have been output by Algorithm 2 instead of \tilde{p} because an equality holds but contrary to \tilde{p} , there exists for p a non zero service rate from N_a to N_b with $Q_{ab}(t) > 0$ and $Q_b(t) \leq Q_b^{\text{lim}}(t)$. \square

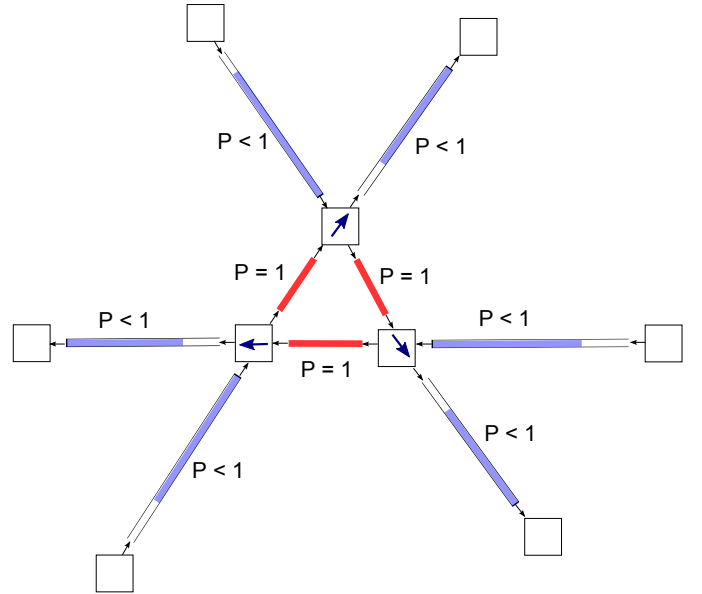


Fig. 8. Work conservation and deadlock resolution using normalized pressure functions. The pressure at congested nodes equals 1, while it is strictly lower than 1 at non-congested nodes. As a result, the phase computed by Algorithm 2 is in favor of congested nodes emptying.

One can expect the work-conservation property to improve the efficiency of back-pressure control in terms of stability because servers at junctions are more likely to work, even in congested conditions. The next section presents simulation results which confirm this expectation, and performance is increased by almost 50% in the particular conditions of the presented simulations.

V. SIMULATION RESULTS

A. The simulation platform

The model and the algorithms presented in this paper have been implemented into a simulator coded in Java.

- The simulator simulates a grid network, as the one depicted in Figure 9. Every junction has 4 inputs, 4 outputs, and 4 feasible phases as depicted in Figure 2.

- The capacity of the nodes can be set as desired. For the presented simulations, all nodes have a capacity of 120 vehicles, except at three regions where capacities equal 40, as depicted in Figure 9. This enables to observe the effect of non uniform capacities.
- Every individual flow allowed by phases of Figure 2 equals 10, and the slot size is 10, so that $\Delta Q_a^{\max} = 10$ for all nodes.
- We delay transfers from node to node with the most simple model: a delay proportional to the distance to reach the queue. This delay shifts queues up, so that the effect of congestion is easier to observe.
- Vehicles are generated at every node with the same arrival rate which can be set as desired. The arrival process generates individual arrivals as well as batches of 10 vehicles. Then, their routing decision at every junction is probabilistic as well as the number of travelled nodes.

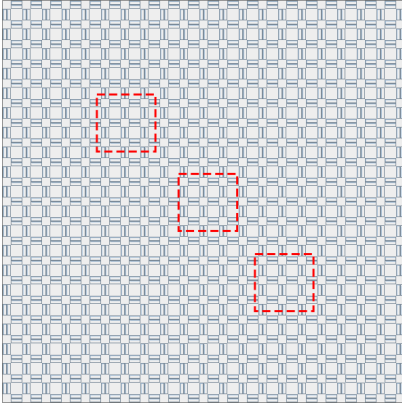


Fig. 9. The 21×21 grid network used for the presented simulations. Nodes in the dashed surrounded regions have a capacity of 40. Other nodes have a capacity of 120 vehicles.

B. Evaluation of the capacity-aware back-pressure control

Simulations have been carried out for the grid network of Figure 9, composed of three regions of smaller capacities, with the following parameters:

- Turn left probability at a junction: 0.1
- Turn right probability at a junction: 0.1
- Probability of a batch: 0.05
- Pressure functions of Equation 20 with $m = 2$ and $C_\infty = 500$.

Vehicles are generated with a constant arrival rate during 1500 time slots, then arrivals stop and there are only endogenous flows and exits. Three experiments are carried out at four different uniform arrival rates over nodes: 0.2, 0.25, 0.3 and 0.35 vehicles per time slot. For each arrival rate, simulations are run 10 times with back-pressure control under linear pressure functions and 10 times with back-pressure control under normalized pressure functions.

At arrival rate 0.2, both controls are efficient and at the end of the simulations, the queuing network is empty. For arrival rates 0.25 and 0.3, the queuing network gets unstable over the simulation when back-pressure control under linear

pressure functions is applied. At the end of the simulation, congestion has propagated and the system is stuck in a deadlock. In contrast, if back-pressure under normalized pressure functions is applied, the queuing network is stable over the ten simulations and the network gets empty at the end, as depicted in Figure 10. However, note that it is very likely that more simulation runs would have enabled to observe that even using normalized pressures, the queuing network can get unstable and stuck in a deadlock over the simulation, because it is a non zero probability event. The key benefit of normalized pressure functions, as simulations tend to indicate, is that its probability of occurrence is strongly decreased. Figure 11 depicts the global queue of the network over time, i.e. $\sum_a Q_a$, for one of the simulations runs with an arrival rate of 0.3 vehicles per time slot under both linear/normalized pressure functions. For an arrival rate of 0.35 vehicles per time slot, back-pressure under both linear/normalized pressure are unable to mitigate congestion propagation. In conclusion, for the particular case of the network used for the presented simulations, the performance in terms of "statistical stability" has been increased by almost 50%.

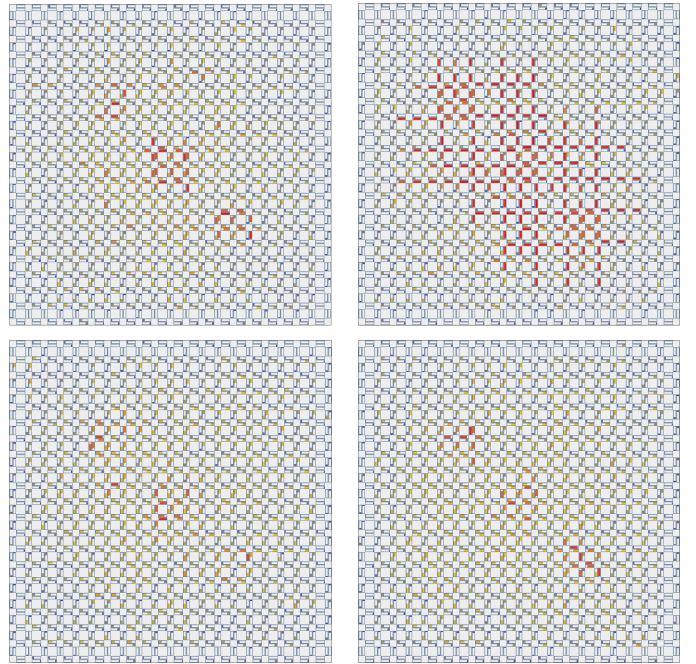


Fig. 10. Evolution of the queuing network using back-pressure control under linear pressure functions (top plots) and normalized convex pressure functions (bottom plots) with an exogenous arrival rate of 0.3 vehicles per time slot. The left (resp. right) drawing depicts the state of the network after 200 (resp. 400) time slots. Under linear pressure functions, congestion propagates very fast and the queuing network gets stuck in a deadlock. Under normalized pressure functions, congestion propagation is mitigated, and the queuing network will eventually completely empty when arrivals will stop.

VI. CONCLUSIONS AND PERSPECTIVES

In this paper, we propose to adapt state-of-the-art back-pressure control to take into account bounded queues constraints. Non work-conservation of current back-pressure control is proved, and identified as a source of congestion propagation through the network. This phenomenon is caused by

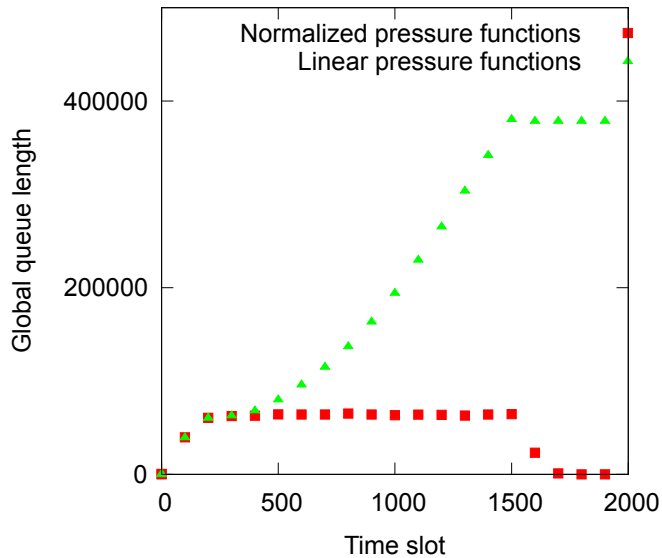


Fig. 11. Evolution of the global queue length over time with an exogenous arrivals rate of 0.3 vehicles per time slot using back-pressure control under linear pressure functions (triangles), and normalized pressure functions (squares). Under linear pressure functions, the queuing network gets unstable very fast and the global queue length does not go back to zero after arrivals stop, because the network is stuck in a strong deadlock due to congestion propagation. Under normalized pressure functions, the queuing network is stable over the simulation, it eventually gets empty after several time slots when arrivals stop.

pressure saturation at queues that have reached maximum capacity.

Normalized pressure functions are proved to ensure work-conservation and this property tends to indicate that congestion propagation will be mitigated. Simulations confirm the efficiency of the approach. It is remarkable that performance have been increased under bounded queues constraints as indicated by simulations, while the ability to distribute the control over junctions and $\mathcal{O}(1)$ complexity properties have been conserved.

However, for very high arrival rates, and in particular above the capacity region, congestion will necessarily eventually propagate through the network. In this case, vast areas of the network will be congested and inter-junctions interactions due to blocking tend to indicate that phase control should be carried out on groups of junctions belonging to the same congested region. Nevertheless, this task is of high complexity due to the exponential complexity of inter-junctions interactions.

Finally, future works on back-pressure signal control should consider the feedback loop between traffic signal control and driver behaviour, and in particular driver routing choice. One can expect drivers at a junction to change their routing choice if the traffic light gives the right of way in favour of some particular output nodes due to traffic conditions, and in particular due to congestion at some nodes. It is of high interest to take into account such behaviours, since they may stabilize or unstabilize the queuing network.

REFERENCES

- [1] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *Journal of Artificial Intelligence Research*, vol. 31, pp. 591–656, March 2008.
- [2] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, "Review of road traffic control strategies," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2043–2067, 2003.
- [3] P. Hunt, D. Robertson, R. Bretherton, and M. Royle, "The scoot on-line traffic signal optimisation technique," *Traffic Engineering & Control*, vol. 23, no. 4, 1982.
- [4] P. Lowrie, "Scats, sydney co-ordinated adaptive traffic system: A traffic responsive method of controlling urban traffic," 1990.
- [5] J.-J. Henry, J.-L. Farges, and J. Tuffal, "The prodyn real time traffic algorithm," in *Proceedings of the 4th IFAC/IFORS Conference on Control in Transportation Systems*, 1984.
- [6] P. Mirchandani and L. Head, "A real-time traffic signal control system: architecture, algorithms, and analysis," *Transportation Research Part C: Emerging Technologies*, vol. 9, no. 6, pp. 415–432, 2001.
- [7] N. H. Gartner, "Opac: A demand-responsive strategy for traffic signal control," *Transportation Research Record*, no. 906, 1983.
- [8] C. Diakaki, M. Papageorgiou, and K. Aboudolas, "A multivariable regulator approach to traffic-responsive network-wide signal control," *Control Engineering Practice*, vol. 10, no. 2, pp. 183–195, 2002.
- [9] C. Osorio and M. Bierlaire, "A surrogate model for traffic optimization of congested networks: an analytic queueing network approach," *Report TRANSP-OR*, vol. 90825, pp. 1–23, 2009.
- [10] H. K. Lo, E. Chang, and Y. C. Chan, "Dynamic network traffic control," *Transportation Research Part A: Policy and Practice*, vol. 35, no. 8, pp. 721–744, 2001.
- [11] K. Aboudolas, M. Papageorgiou, and E. Kosmatopoulos, "Store-and-forward based methods for the signal control problem in large-scale congested urban road networks," *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 2, pp. 163–174, 2009.
- [12] A. Di Febbraro, D. Giglio, and N. Sacco, "On applying petri nets to determine optimal offsets for coordinated traffic light timings," in *Intelligent Transportation Systems, 2002. Proceedings. The IEEE 5th International Conference on*, pp. 773–778, 2002.
- [13] S. Shepherd, "A review of traffic signal control," 1992.
- [14] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *Automatic Control, IEEE Transactions on*, vol. 37, no. 12, pp. 1936–1948, 1992.
- [15] P. Varaiya, "The max-pressure controller for arbitrary networks of signalized intersections," in *Advances in Dynamic Network Modeling in Complex Transportation Systems*, pp. 27–66, Springer, 2013.
- [16] P. Varaiya, "A universal feedback control policy for arbitrary networks of signalized intersections," tech. rep., 2009.
- [17] T. Wongpiromsarn, T. Uthacharoenpong, Y. Wang, E. Frazzoli, and D. Wang, "Distributed traffic signal control for maximum network throughput," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pp. 588–595, IEEE, 2012.
- [18] M. J. Neely, *Dynamic power allocation and routing for satellite and wireless networks with time varying channels*. PhD thesis, LIDS, Massachusetts Institute of Technology, 2003.
- [19] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 23, no. 1, pp. 89–103, 2005.
- [20] H. G. Perros, *Queueing networks with blocking*. Oxford University Press, Inc., 1994.
- [21] S. Kundu and I. F. Akyildiz, "Deadlock free buffer allocation in closed queueing networks," *Queueing Systems*, vol. 4, no. 1, pp. 47–56, 1989.
- [22] J. Gregoire, S. Bonnabel, and A. De La Fortelle, "Priority-based coordination of robots." 29 pages, June 2013.
- [23] P. Giaccone, E. Leonardi, and D. Shah, "Throughput region of finite-buffered networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 18, no. 2, pp. 251–263, 2007.