Traffic Signal Control Methods

David Yao

2015/05/02

1 The Traffic Signal Control Problem

1.1 Introduction

Urban traffic networks are facing serious congestion problems, in part due to sub-optimal control policies governing the traffic lights. My report will look at and compare several decentralized, dynamic algorithms recently proposed in the literature, all based on the "Back Pressure" algorithm from [1].

1.2 Model and Terminologies

Here I give a very general model of the traffic signal control problem. Consider a network of intersecting roads $\mathcal{N}$ as a tuple $\mathcal{N} = (\mathcal{L}, \mathcal{J})$ where $\mathcal{L} = \{L_1, \ldots, L_N\}$ is the set of road segments with direction (called **links**) so that a two-way street segment gives two links in $\mathcal{L}$, and $\mathcal{J} = \{J_1, \ldots, J_L\}$ is the set of nodes of intersection, called **junctions**. For each node $n \in \mathcal{J}$, $I(n) \subset \mathcal{L}$ is the set of links entering n, and $O(n) \subset \mathcal{L}$ is the set of links leaving n. Since all links are uni-directional, assume $\forall n \in \mathcal{J}, I(n) \cap O(n) = \emptyset$. $M_n \subset I(n) \times O(n)$ is the set of all possible movements through n. $\mathcal{S}_n \subset \mathcal{P}(M_n)$ is a collection of subsets of M_n, whose elements are sets of simultaneously permissible movements through n. Note that M_n's are all disjoint for $n \in \mathcal{J}$. Let $\mathcal{M} = \bigcup_{n \in \mathcal{J}} M_n$ be the union of these possible movements at all junctions. My terminology will follow the European Standard where elements of M_n are called **phases** or **movements**, and elements of $\mathcal{S}_n$ is called **stages**. See Figure 1 for an example of a junction and its allowable stages.

Our model operates on discrete time slots of τ seconds, which is the minimum duration for any stage at a junction. Time slots are labelled by $t \in \mathbb{N}$. Furthermore, for $l \in \mathcal{L}$, let $In_l \subset \mathcal{L}$ be the set of links that can go into l, i.e. $In_l \subset I(n)$ where n is the unique junction s.t. $l \in O(n)$ unless l leads traffic from outside the network into the network, in which case $In_l = \emptyset$. Similarly define Out_l. For each link $l \in \mathcal{L}$, let $Q_l(t)$ be the number of vehicles queueing at link l at the end of time slot t. For each phase $(a, b) \in \mathcal{M}$, let $r(a, b)$ be the proportion of vehicles entering b upon leaving a, and let $c(a, b)$ be the saturation flow for this phase, in vehicles per time slot. Finally, vehicles can enter and exit the network at any link, let $D_l(t)$ be the number of exogenous vehicles entering link l at time slot t. Let $d_l = \mathbb{E}[D_l(t)]$ be the expected arrival rate in link l. The exit proportion for link l is just $1 - \sum_{m \in Out_l} r(l, m)$.

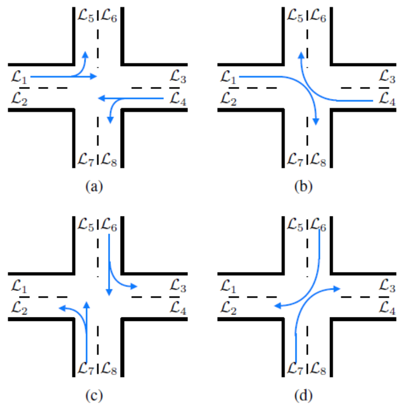

Figure 1: Typical $\mathcal{S}_n = \{S_a, S_b, S_c, S_d\}$ for a 4-way junction n where $I(n) = \{\mathcal{L}_1, \mathcal{L}_4, \mathcal{L}_6, \mathcal{L}_7\}$, $O(n) = \{\mathcal{L}_2, \mathcal{L}_3, \mathcal{L}_5, \mathcal{L}_8\}$
(a) $S_a = \{(\mathcal{L}_1, \mathcal{L}_3), (\mathcal{L}_1, \mathcal{L}_5), (\mathcal{L}_4, \mathcal{L}_2), (\mathcal{L}_4, \mathcal{L}_8)\}$
(b) $S_b = \{(\mathcal{L}_1, \mathcal{L}_8), (\mathcal{L}_4, \mathcal{L}_5)\}$
(c) $S_c = \{(\mathcal{L}_6, \mathcal{L}_3), (\mathcal{L}_6, \mathcal{L}_8), (\mathcal{L}_7, \mathcal{L}_2), (\mathcal{L}_7, \mathcal{L}_5)\}$
(d) $S_d = \{(\mathcal{L}_6, \mathcal{L}_2), (\mathcal{L}_7, \mathcal{L}_3)\}$

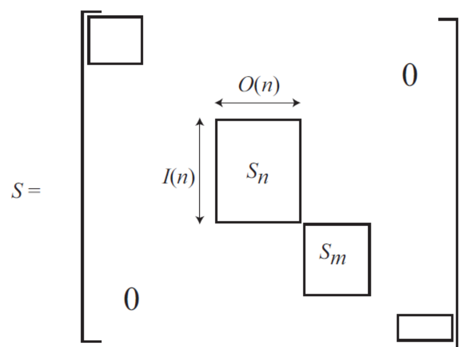

Figure 2: Control Matrix S as a block diagonal matrix with junction control matrices S_n, S_m along the diagonal

The objective is to select for each time slot, $(S_n(t) \in \mathcal{S}_n,\ n \in \mathcal{J})$ for $t = 1, 2, 3, \ldots$, so that all queues $(Q_l(t),\ l \in \mathcal{L})$ are stable, or equivalently, that the long term average of all queue lengths is bounded, or that the Markov Chain $(Q_l,\ l \in \mathcal{L})$ is positive recurrent.

Note: M_n are all disjoint, and for each link l, there is at most one n s.t. $l \in I(n)$, and at most one m s.t. $l \in O(m)$. According to [2], $S_n(t)$ can be represented by a binary matrix where $S_n(t)(m, l) = 1$ if and only if $(m, l) \in S_n(t)$. Furthermore, $S(t) = \bigcup_{n \in \mathcal{J}} S_n(t)$ can be represented by a single block diagonal matrix $S(t)$ with $S_n(t)$ as diagonal blocks, see Figure 2. Therefore the control policy essentially selects $S(t)$ for each t. Let $\mathcal{S}$ be the set of all possible selections of $S(t)$. It is the Cartesian product $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \times \mathcal{S}_L$ of all possible stages for each junction.

2 Refresher of the Back Pressure Algorithm

2.1 Back Pressure Algorithm

I will first recall the back pressure algorithm. In the Multihop Radio Network considered in [1], a network with N nodes and L links represented by a directed graph $\mathcal{N} = (\mathcal{L}, \mathcal{J})$ are considered. $\mathcal{L}$ is the set of nodes and $\mathcal{J}$ is the set of links. Customers come in J different classes, each with a different set of destination nodes. Customers may come at any node. Time is discreet and at each time slot, only certain subset of the links can serve customers, and each server can serve one class of customers. Let S be the set of subsets of $\mathcal{J}$ that can be served simultaneously. Link i serves the customer successfully with probability m_i from node $q(i)$ to node $h(i)$. Customers do not commit to an outgoing link when they arrive. Let $Q_{aj}(t)$ be the length of the virtual queue at node a for customers of class j. Then the back pressure algorithm, which achieves maximum throughput, finds the links and customer classes to serve in 3 stages.

Stage 1 For each link i, a weight $W_i(t)$ is computed by

$$W_{ij}(t) = \begin{cases} (Q_{q(i)j}(t-1) - Q_{h(i)j}(t-1))m_i & \text{if } h(i) \text{ is not one of the destinations of class } j \\ Q_{q(i)j}(t-1) & \text{if otherwise} \end{cases}$$

Then $W_i(t) = W_{i\hat{j}_i}(t)$ is the maximum across all W_{ij}'s, $j = 1, 2, \ldots, J$.

Stage 2 The activation set $\hat{c}$ is selected by

$$\hat{c} = \arg\max_{c \in S} \sum_{i \in c} W_i(t)$$

Stage 3 For each link i, activate it at slot t if and only if $i \in \hat{c}$ and let it serve class $\hat{j}_i$.

2.2 Motivation for Considering the Back Pressure Algorithm

Our model for the Traffic Control Problem (TCP) is similar to that of the Multihop Radio Network (MRN) in many ways, as I have suggested by using the same symbols. The phases are the links in MRN, each junction in TCP corresponds to a collection of the links in MRN and the links (roads with directions) in TCP are the nodes in MRN. Choosing a stage for each junction is like choosing a class to serve for each link while making sure that certain links (certain phases) are not activated simultaneously. However there are several differences too, as pointed out in [3].

Firstly the final destination of each vehicle is unknown to the TCP controller. Thus instead of dividing the vehicles into classes based on their exit links, the best one can do is to only partition the vehicles queueing at each link according to the immediate outgoing link at each junction. This induces another difference: in MRN the customers do not commit to an outgoing link until a routing decision is made, but in TCP the vehicles are sorted into different lanes for different outgoing links before their phase is activated.

Secondly in MRN the virtual queues are easily created, but in TCP one lane can have vehicles that go to multiple links and thus makes it impossible to fully divide the vehicles into finer queues by the link they are going into. See Figure 1 for an example: the left-most lane on $\mathcal{L}_1$ can enqueue both vehicles going into $\mathcal{L}_3$ and $\mathcal{L}_5$, since these two phases are always simultaneously activated in the allowable stages. Moreover, as [4] points out, even when each lane corresponds to only one outgoing link, vehicles do not split up by lanes until they are close to the junction. Nevertheless, the ability to measure these finer queues is an important assumption made by the algorithm proposed by [2].

Thirdly in MRN the controller has complete control over the routing of the customers, but in TCP, the controller only actuates stages and if during a certain stage, a vehicle can go to multiple outgoing links, the routing is the driver's choice. Fortunately, the ratios $r(a, b)$ can help characterize the overall behavior of large number of vehicles.

Despite the differences, the idea of using back pressure weighted by service rates still provides an effective way of designing a distributed traffic control system.

3 Universal Feedback Control Policy (UCP) by Varaiya(2009)

3.1 Additional Assumptions

The links $\mathcal{L}$ in UCP is a special case of the general model, where $\mathcal{L}$ is partitioned into three subsets $\mathcal{L}_{internal}$, $\mathcal{L}_{entry}$ and $\mathcal{L}_{exit}$. d_l is non-zero only for link $l \in \mathcal{L}_{entry}$, exit proportion is 1 for all $l \in \mathcal{L}_{exit}$, and exit proportion is 0 for all other links.

UCP also assumes the controller has the finer queue lengths $Q_{ab}(t)$ for each time slot t, where $Q_{ab}(t)$ denotes the number of vehicles queueing on link a that intend to turn to link b. Naturally $\forall a \in \mathcal{L},\ Q_a(t) = \sum_{b \in Out_a} Q_{ab}(t)$ in this model, since exit links do not lead to any other links.

3.2 Characterizing the Throughput Region

First let f_l be the flow of vehicles on link l. It can be shown that for every demand $d = (d_l, l \in \mathcal{L}_{entry})$, there is a unique flow $f = f(d) = (f_l, l \in \mathcal{L})$ satisfying:

$$f_l = \begin{cases} d_l, & l \in \mathcal{L}_{entry} \\ \sum_{m \in In_l} f_m r(m, l), & l \in \mathcal{L} \setminus \mathcal{L}_{entry} \end{cases}$$

Then by using the family of randomize policies, [2] shows that the throughput region D is the set of all d such that for the corresponding flow $f = f(d)$, $\exists \Sigma \in co(\mathcal{S})$ (convex hull of the set of control matrices) s.t.

$$\forall (l, m) \in M, f_l r(l, m) \le c(l, m)\Sigma(l, m)$$

3.3 The Control Algorithm

The algorithm is distributed: to compute the stage $S_n(t)$ for junction n, it only require information about links near n, and computes $S_n(t)$ in three stages.

Stage 1 For each junction n and each phase (l, m) in M_n, a (back-pressure-like) weight $W_{n,(l,m)}(t)$ is computed by

$$W_{n,(l,m)}(t) = Q_{lm}(t-1) - \sum_{p \in Out_m} r(m,p) Q_{mp}(t-1)$$

Stage 2 Calculate the gain $\gamma_n(S_n)(t)$ for each possible stage $S_n \in \mathcal{S}_n$ by

$$\gamma_n(S_n)(t) = \sum_{(l,m) \in S_n} c(l,m) W_{n,(l,m)}(t)$$

Stage 3 Select the stage $S_n(t) = \hat{S_n}$ to activate by

$$\hat{S_n} = \arg\max_{S_n \in \mathcal{S}_n} \gamma_n(S_n)(t)$$

3.4 Evaluation

As [2] shows, UCP achieves the maximum throughput as characterized in previous section. Although it uses a special link set $\mathcal{L}$, it is shown in [4] that it is generalizable to network structures of our general model with little modification. However, there are still several limitations. Firstly, as pointed out earlier, UCP requires knowledge of $r(l, m)$ for all $(l, m) \in M$, which is hard to come by accurately in practice. Secondly, it requires lengths of the finer queues $Q_{ab}(t)$ for all $(a, b) \in M$, which, as discussed before, are hard to measure accurately as well. Thirdly, in practice it is possible for congestion at one link l to extend to affect all links $m \in In_l$. UCP, however, assumes infinite buffer capacities for all links. It is not quite clear whether UCP is still stabilizing in the model where links have limited buffer size.

4 Another distributed algorithm by Wongpiromsarn et al. (2012)

4.1 Additional Assumptions

In contrast with UCP discussed in previous section, [3] does not assume finer queue lengths $Q_{ab}(t)$ or routing ratios $r(a, b)$. However, for each junction n, it requires a finite set Z_n of traffic states around junction n and a rate function $\xi_n : \mathcal{S}_n \times M_n \times Z_n \to \mathbb{N}_0$ that, given the current stage, movement and traffic state, gives the rate of traffic flow. It also assumes that the vector of traffic states $z(t) = (z_n(t), n = 1, 2, 3, \dots, L)$ is a finite state, irreducible aperiodic Markov Chain with a stationary distribution. Let $\mathcal{Z} = Z_1 \times \cdots \times Z_L$

4.2 Characterize the Throughput Region

Define the vector function $\xi : \mathcal{S} \times \mathcal{Z} \to \mathbb{N}_0^{|M|}$ where the kth coordinate of $\xi(S, z)$ is $\xi_n(S_n, (a, b), z_n)$ where (a, b) is the kth phase in M and n is the unique junction s.t. $a \in I(n)$ and $b \in O(n)$, denote this kth coordinate by $\xi(S, z)_{ab}$. Suppose the stationary distribution of $z(t)$ is $\mathbb{P}[z(t) = z] = \pi_z$. Now define:

$$\Gamma = \sum_{z \in \mathcal{Z}} \pi_z co\{\xi(S, z) | S \in \mathcal{S}\}$$

Then the throughput region D is the set of all d such that $\exists \Sigma \in \Gamma$ together with a flow $f : M \to \mathbb{R}$ that satisfies:

$$f(l,m) \geq 0, \forall (l,m) \in M$$

$$d_l = \sum_{m \in Out_l} f(l,m) - \sum_{p \in In_l} f(p,l), \forall l \in \mathcal{L}$$

$$f(l,m) = \Sigma_{lm}, \forall (l,m) \in M$$

4.3 The Control Algorithm

The algorithm is also distributed, and is similar to UCP. It computes $S_n(t)$ in three stages:

Stage 1 For each junction n, and each phase (l,m) in M_n, the back-pressure weight $W_{n,(l,m)}(t)$ is computed by:

$$W_{n,(l,m)}(t) = Q_l(t-1) - Q_m(t-1)$$

Stage 2 Calculate the gain $\gamma_n(S_n)(t)$ for each possible stage $S_n \in \mathcal{S}_n$ by:

$$\gamma_n(S_n)(t) = \sum_{(l,m) \in S_n} \xi_n(S_n, l, m, z_n(t)) W_{n,(l,m)}(t)$$

Stage 3 Select the stage $S_n(t) = \hat{S_n}$ to activate by:

$$\hat{S_n} = \underset{S_n \in \mathcal{S}_n}{\arg\max}\, \gamma_n(S_n)(t)$$

4.4 Evaluation

[3] shows that the algorithm achieves maximum throughput provided that $z(t)$ is i.i.d. from slot to slot. It seems to use only the aggregated queue lengths $Q_a(t)$ for each link a and it does not seem to require the routing ratio $r(a,b)$ either. However, the additional assumption of traffic state space $\mathcal{Z}$ is in fact so strong that it encodes almost all the assumptions of UCP.

Just using aggregated queue length $Q_a(t)$ is not enough for UCP because the controller would not know which outgoing link requires most service for vehicles on link a. Comparing only aggregated queue lengths can lead the system to activate movements from a that do not need service, and as a consequence waste resource. However, ξ_n can presumably compensate for that because in this particular traffic state where $Q_{ab}(t) = 0$, $\xi_n(S_n, a, b, z_n(t)) = 0$ as well, so the weight $W_{n,(a,b)}(t)$ does not contribute to the stage S_n.

$\xi_n(S_n, a, b, z_n(t))$ also implicitly uses the routing ratio $r(a,b)$ when $z_n(t)$ represent the cases of saturated flow. The rate at which vehicles go from a to b in this situation should be $c(a,b)r(a,b)$, where $c(a,b)$, used in UCP, is the saturated flow assuming all traffic on a is going into b.

In summary, although the back-pressure weight is computed using only the aggregated queue length, the traffic state $z(t)$ in this algorithm encodes a great amount of information and as a result, $\mathcal{Z}$ can be a very large set. Moreover, the assumption that $z(t)$ is i.i.d. is an over-simplification because each stage choice at a time slot will affect the traffic state at the next slot, and it is not clear whether this algorithm is still achieves optimal throughput when one considers the additional complexity of the traffic states. It is also unknown how the algorithm behaves when links have finite buffer sizes.

5 Variation on UCP by Gregoire et al. (2014)

5.1 Additional Assumptions

The algorithm proposed in [4], called BP Control, assumes neither finer queue lengths nor routing ratios. It only requires a vehicle detector variable in $[0, 1]$ defined as $\delta_{ab}(t) = \min(\frac{Q_{ab}(t)}{c(a,b)}, 1)$. It signifies a degree to which the finer queue $Q_{ab}(t)$ is non-empty.

5.2 Characterizing the Throughput Region

Since BP is a variation of UCP with no change to the model, the throughput region is the same as the one D defined in Section 3.2.

5.3 The Control Algorithm

BP is again distributed and computes $S_n(t)$ in three stages.

Stage 1 Foe each junction n and each phase $(l, m) \in M_n$, a back-pressure weight $W_{n,(l,m)}(t)$ is computed by

$$W_{n,(l,m)}(t) = \delta_{lm}(t)max(Q_l(t-1) - Q_m(t-1), 0)$$

Stage 2 Calculate the gain $\gamma_n(S_n)(t)$ for each possible stage $S_n \in \mathcal{S}_n$ for junction n by

$$\gamma_n(S_n)(t) = \sum_{(l,m)\in S_n} c(l, m)W_{n,(l,m)}(t)$$

Stage 3 Select the stage $S_n(t) = \hat{S_n}$ to activate by

$$\hat{S_n} = \arg\max_{S_n\in\mathcal{S}_n} \gamma_n(S_n)(t)$$

5.4 Evaluation

The BP Controller only uses quantities that are relatively easy to measure accurately, such as aggregated queue length $Q_a(t)$ and vehicle detector variable $\delta_{ab}(t)$, which can be measured by loop detectors positioned at lanes dedicated to one specific movement. However, it no longer achieves maximum throughput. Characterizing the stability region of BP is still a challenging problem, but [4] compared the performance of BP and UCP by simulations on a 21×21 grid network with 4-way junctions of the kind shown in Figure 1. The relevant parameters are randomized, and BP on averaged supported around 80% of the maximum arrival rates supported by UCP. However, other network topologies need to be investigated to further support the claim that BP achieves a significant part of the throughput region. However, finite buffer sizes for links are still not considered.

6 Conclusion

The approaches of both UCP by [2] and the algorithm by [3] achieve maximum throughput of the network, though with different models. UCP requires at each time slot the number of vehicles queueing at each link for each outgoing link, as well as the knowledge of routing ratios. The latter algorithm, on the other hand, requires a rate function on a finite traffic state space which potentially can be very large and, for the most part, encapsulates the two requirements of UCP. The BP algorithm in [4] foregoes some throughput performance in order to eliminate these two assumptions of UCP and gives a more realistic and practical solution. The loss of performance is only roughly 20% for grid networks but its performance for arbitrary network topologies need to be further investigated. None of the three algorithms considered in the report deals with the possibility that links have finite buffer sizes.

References

[1] Leandros Tassiulas and Anthony Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions On Automatic Control*, 37(12):1936–1948, December 1992.

[2] Pravin Varaiya and Topl Group. A universal feedback control policy for arbitrary networks of signalized intersections, September 2009.

[3] Tichakorn Wongpiromsarn, Tawit Uthaicharoenpong, Yu Wang, Emilio Frazzoli, and Danwei Wang. Distributed traffic signal control for maximum network throughput. In *15th International IEEE Conference on Intelligent Transportation Systems*, pages 588–595, Hilton Anchorage, Anchorage, AK, USA, September 2012.

[4] Jean Gregoire, Emilio Frazzoli, Arnaud de La Fortelle, and Tichakorn Wongpiromsarn. Back-pressure traffic signal control with unknown routing rates. In *Proceedings of the 19th IFAC World Congress*, volume 19, pages 11332–11337, Cape Town International Convention Centre, Cape Town, South Africa, August 2014. International Federation of Automatic Control.