53rd IEEE Conference on Decision and Control
December 15-17, 2014. Los Angeles, California, USA

Network–decentralized robust congestion control with node traffic splitting

Franco Blanchini [a], Giulia Giordano [a], Pier Luca Montessoro [b]

***Abstract*— We consider a traffic control problem defined on a network graph, whose nodes represent buffers and whose arcs represent flow channels. We consider network models with a peculiar aspect: each element of the flow arriving at each node must be redirected towards a precise other node of the network, hence each buffer is naturally split in several queues, characterized according to statistics about the flow splitting at the nodes. Precisely, each node is modelled as a Markov chain, in which some states are specifically associated with the arcs leaving the node: state j represents the amount of traffic waiting to be directed through arc j. We show that such a network can be stabilized by means of a network–decentralized control, in which the flow through each arc is controlled by an agent which only knows the congestion situation at the nodes it connects.**

The main result is that the proposed network–decentralized strategy is robust (namely it assures stability under all possible values of the Markov chain parameters) provided that zero is a simple eigenvalue for all the Markov chains, which includes the irreducible case.

I. Introduction and Motivation

Traffic and congestion control is fundamental in many different applications: vehicle congestion on highways [12], [13], large data communication networks [11], [14], [15], [17], [18], inventory management and production–distribution systems [5], [6], [9], [10], [21], [22], water distribution networks [3], [16], transportation networks [2], [19] and network flows in general [1], [4], [20], [23].

Typically, a flow control problem is formalized on a network in which the nodes represent buffers, while the arcs represent flow channels. The flow through an arc can be either controlled or non–controllable. One fundamental issue in network control is that the traffic at each node is formed by elements which have to be forwarded towards different directions. This is typical in traffic and data networks, in which buffers include packets or vehicles which have to be routed towards different adjacent nodes. Thus, an appropriate modelling framework should take into account the different queues populating each node: each queue is related to a stream with a different direction.

The main problem is that the partition of the node traffic in queues, associated with different directions, is not a controlled variable. Here we assume that, for each node, a statistical distribution is available about the exit directions of the node population. We model this statistics as a continuous–time Markov matrix, which corresponds to a Metzler matrix with zero–sum columns. We seek linear *network–decentralized* [3], [9], [13], [14], [15] state–feedback controllers, such that each control agent (*arc*) can use information only from the subsystems (*nodes*) it connects. This is equivalent to imposing that the state feedback matrix has the same structure as the transpose of the input matrix.

In [7], [8], a network–decentralized control strategy has been proposed which can be applied when the nodes are arbitrary subsystems (with their own, possibly unstable, dynamics) and each control agent knows the states of the nodes it directly affects. Such a control strategy requires the knowledge of the node dynamics as well: this may be a problem, since the coefficients of the Markov chains are likely to be uncertain and can even depend on external factors. Here we present a network–decentralized strategy which is robust: agents do not need to know exactly the situation of each of the nodes they affect and can simply rely on cumulative information about the total congestion at each node. The main contributions of this paper are summarized next.

- We formulate the problem of node traffic splitting, in which each node is represented by a Markov matrix, and we show to what extent the theory developed in [7], [8] applies.
- As a first main result, if zero is a simple eigenvalue for all the Markov chains (which is always true if they are irreducible), we propose a network–decentralized control strategy which is robust: it assures stability regardless of the coefficients of the Markov chain. Indeed, the control agents do not need to know such coefficients.
- As a second main result, we show that each agent needs to have information only about the total amount of congestion in each of the nodes it influences. It can thus ignore the actual distribution at the nodes.
- We show that such a stabilizing control is effective even in the presence of flow constraints, so extending the results in [3].
- We discuss the case in which zero is a multiple eigenvalue for at least one Markov chain.

A. Modelling node dynamics

Consider a large network in which the nodes represent buffers and the arcs represent flow channels connecting the buffers. In Fig. 1 we sketch a portion of a typical network. Each connection between two nodes, including one or two arcs denoted by u_k, is controlled by an agent (we will denote by v_h). The presence of double arcs indicates that the same

[a] Dipartimento di Matematica e Informatica, Università degli Studi di Udine, 33100 Udine, Italy. blanchini@uniud.it, giulia.giordano@uniud.it

[b] Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica, Università degli Studi di Udine, 33100 Udine, Italy. montessoro@uniud.it

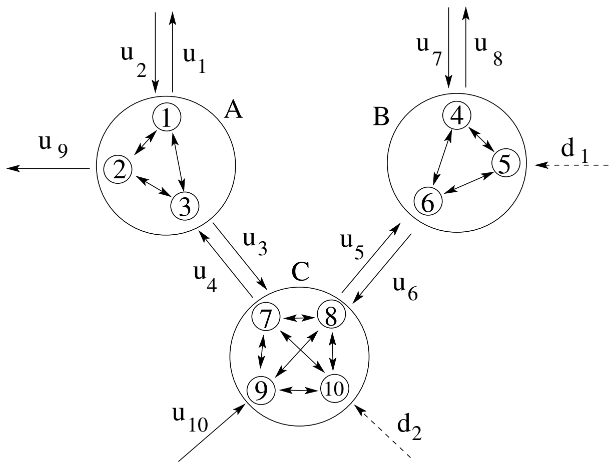

Fig. 1: An example of network with node dynamics.

agent can control the flow in the two opposite directions. Arcs denoted by d_j represent an exogenously determined flow, such as information packets introduced into a routing system or vehicles entering a viability system.

For example, in Fig. 1 there are three nodes, denoted by A, B and C. At each node, the traffic is split in queues having different destinations. In node A, for instance, the different queues are represented by subnodes 1 (traffic directed to the north), 2 (traffic directed to south–east) and 3 (traffic directed to the west).

We model traffic splitting among different directions as a dynamic process, represented by a stochastic matrix. For instance, node A in Fig. 1 could be represented by the matrix

$$M_A = \begin{bmatrix} -(\alpha+\beta) & \gamma & \epsilon \\ \alpha & -(\gamma+\delta) & \phi \\ \beta & \delta & -(\epsilon+\phi) \end{bmatrix},$$

which means that each unit arriving at subnode 1 is directed either to subnode 2 or to subnode 3, with splitting rates $\alpha > 0$ and $\beta > 0$ respectively, and so on. If we assume, for the moment being, that $\gamma = \delta = \epsilon = \phi = 0$, then all the units at subnode 1 are transferred to subnodes 2 and 3. If we consider $x_1(0) = \xi$, $x_2(0) = x_3(0) = 0$, and no external arc flow, asymptotically we have

$$[x_1(\infty) \quad x_2(\infty) \quad x_2(\infty)] = \frac{\xi}{\alpha+\beta} \; [0 \quad \alpha \quad \beta].$$

The arc flows u_9 and u_3 redirect to other nodes the units in subnodes 2 and 3. Note that we admit that the transfer process is not instantaneous but exponential, with mode $e^{-(\alpha+\beta)t}$. Then the magnitude of $\alpha+\beta$ is associated with the process speed and the relative magnitudes $\alpha/(\alpha+\beta)$ and $\beta/(\alpha+\beta)$ are associated with the traffic distribution.

Of course, by considering positive α, β, γ, δ, ϵ and ϕ, we can model the overall traffic distribution at node A.

II. Model and assumptions

We consider a class of linear, interconnected subsystems:

$$\dot{x}_i(t) = A^{(i)} x_i(t) + \sum_{j \in \mathcal{C}_i} B_{ij} v_j(t) + D_i d(t),$$

where $x_i(t) \in \mathbb{R}^{n_i}$ is the state of the ith buffer or subsystem; $\mathcal{C}_i$ is the set that indexes the control subvectors $v_j(t) \in \mathbb{R}^{m_j}$, $j = 1, \ldots, M$, named *agents*, affecting the ith subsystem; B_{ij} represents the effect of control v_j on the ith subsystem; $d(t)$ is an external signal, bounded in a compact and convex set $\mathcal{D}$, affecting the ith subsystem through matrix D_i. Matrices $A^{(i)}$ are stochastic matrices.

Assumption 1: Matrix $A^{(i)}$ is a Metzler matrix,

$$A^{(i)}_{kj} \geq 0, \quad k \neq j,$$

with zero sum columns:

$$\sum_{k=1}^{n_i} A^{(i)}_{kj} = 0.$$

The last condition can be synthetically written as

$$\bar{1}^\top_{n_i} A^{(i)} = 0,$$

where

$$\bar{1}^\top_{n_i} = \underbrace{[\, 1 \; 1 \; \ldots \; 1 \,]}_{n_i \text{ times}}.$$

The overall system can be written as ([7], [8])

$$\dot{x}(t) = Ax(t) + Bu(t) + Dd(t), \tag{1}$$

where $x(t) \in \mathbb{R}^n$ includes the state variables associated with each subsystem, $u(t) \in \mathbb{R}^m$ is the control vector including all the agents $v_j(t)$, $d(t) \in \mathbb{R}^n$ is the vector representing an external, non–controllable signal affecting the system, D is a generic matrix, while A and B are block–structured: $A \in \mathbb{R}^{n \times n}$ is the block–diagonal matrix

$$A = \text{blockdiag}\{A^{(1)}, \ldots, A^{(i)}, \ldots, A^{(N)}\}, \tag{2}$$

while matrix $B \in \mathbb{R}^{n \times m}$ is a suitably *structured* matrix.

System (1) can be naturally represented by a hypergraph, where the N subsystems are associated with nodes and control agents are associated with hyperarcs. For simplicity, we will speak of graphs and arcs.

Each control agent v_j, $j = 1, \ldots, M$ is a vector in $\mathbb{R}^{m_j}$ associated with a block column of B. Matrix B has a special structure: each column has zero blocks $B_{ij} \in \mathbb{R}^{n_i \times m_j}$ corresponding to all the nodes not directly affected by agent v_j. Formally, since $\mathcal{C}_i$ is the set that indexes the agents directly affecting node i, we have

$$B_{ij} = 0 \;\text{ if and only if }\; j \notin \mathcal{C}_i.$$

Denoting by $\mathcal{N}_j$ the set that indexes the nodes affected by agent j, we also have

$$B_{ij} = 0 \;\text{ if and only if }\; i \notin \mathcal{N}_j.$$

All the block dimensions must be compatible with the block structure of A, namely $\sum_{i=1}^N n_i = n$ and $\sum_{i=1}^M m_i = m$.

Example 1: In the case of Fig. 1, there are 3 nodes (A is labeled as node 1, B as node 2 and C as node 3) and 6 agents: $v_1 = [u_1^\top \; u_2^\top]^\top$, $v_2 = [u_3^\top \; u_4^\top]^\top$, $v_3 = [u_5^\top \; u_6^\top]^\top$, $v_4 = [u_7^\top \; u_8^\top]^\top$, $v_5 = u_9$, $v_6 = u_{10}$. We have $\mathcal{C}_1 = \{1, 2, 5\}$, $\mathcal{C}_2 = \{3, 4\}$, $\mathcal{C}_3 = \{2, 3, 6\}$. The agents control the following nodes: $\mathcal{N}_1 = \{1\}$, $\mathcal{N}_2 = \{1, 3\}$, $\mathcal{N}_3 = \{2, 3\}$, $\mathcal{N}_4 = \{2\}$,

$\mathcal{N}_5 = \{1\}$, $\mathcal{N}_6 = \{3\}$. Matrices B and D are

$$B = \begin{bmatrix} B_{11} & B_{12} & \mathbf{0} & \mathbf{0} & B_{15} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & B_{23} & B_{24} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & B_{32} & B_{33} & \mathbf{0} & \mathbf{0} & B_{36} \end{bmatrix},$$

$$D = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ D_{21} & \mathbf{0} \\ \mathbf{0} & D_{32} \end{bmatrix}.$$

We clearly work under the following assumption.

Assumption 2: (A, B) is stabilizable.

Along the lines in [7], [8], we consider controls restricted to the following class.

Definition 1: A control is *network–decentralized* if each agent v_j can have information from nodes in $\mathcal{N}_j$ only:

$$v_j = \phi(x_i,\ i \in \mathcal{N}_j).$$

In the case of a linear state feedback, a control of the form $u = -Kx$ is *network–decentralized* if K has the same structural zero blocks as $B^\top$. In Example 1, K should have the structure

$$K = \begin{bmatrix} K_{11}^\top & K_{12}^\top & \mathbf{0} & \mathbf{0} & K_{15}^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & K_{23}^\top & K_{24}^\top & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & K_{32}^\top & K_{33}^\top & \mathbf{0} & \mathbf{0} & K_{36}^\top \end{bmatrix}^\top.$$

III. A ROBUST SOLUTION (MAIN RESULT)

Denote by $e_n = \bar{1}_n/\sqrt{n}$ the averaged unit vector with n components:

$$e_n = [\ \frac{1}{\sqrt{n}}\ \frac{1}{\sqrt{n}} \dots \frac{1}{\sqrt{n}}\ ]^\top.$$

As usual, we refer to 0 as the system equilibrium, which in the case of buffers corresponds to the desired set–point.

Assumption 3: The eigenvalue $\lambda = 0$ of $A^{(i)}$ is simple for all i and all other eigenvalues have a strictly negative real part.

Note that the assumption holds if the stochastic matrices $A^{(i)}$ are irreducible[1]. The opposite is not true. For instance,

$$A^{(i)} = \begin{bmatrix} -(\alpha+\beta) & 0 & 0 \\ \alpha & -\gamma & \delta \\ \beta & \gamma & -\delta \end{bmatrix}, \quad \text{with} \quad \alpha, \beta > 0,$$

is a reducible matrix, whose spectrum is $\sigma_{A^{(i)}} = \{-(\alpha+\beta), -(\gamma+\delta), 0\}$. Hence the eigenvalue 0 has multiplicity 2 if $\gamma = \delta = 0$, otherwise it is simple.

In view of Assumption 3, the following proposition is immediate.

Proposition 1: (A, B) is stabilizable iff $\text{rank}[A|B] = n$.

We now consider the candidate control

$$u(t) = -\kappa B^\top H x(t) \tag{3}$$

where $\kappa > 0$ and

$$H = \text{blockdiag}\{e_{n_1}e_{n_1}^\top, \dots, e_{n_i}e_{n_i}^\top, \dots, e_{n_N}e_{n_N}^\top\}. \tag{4}$$

The following theorem is our main result.

[1]a Metzler matrix $\tilde{M}$ is irreducible if there is no variable permutation such that $\tilde{M} = \begin{bmatrix} M_{11} & 0 \\ M_{21} & M_{22} \end{bmatrix}$

Theorem 1: Under Assumptions 2 and 3, the network–decentralized control (3)–(4) robustly stabilizes the system.

Before proving the theorem, we point out some relevant properties of the candidate control (3)–(4).

- **Claim 1:** the control is network–decentralized.
- **Claim 2:** the control is robust, *i.e.* independent of the parameters of the stochastic matrices.
- **Claim 3:** each control agent needs to know only the cumulative buffer content, and not its distribution.

The last claim needs to be further explained. First, notice that $e_{n_i}e_{n_i}^\top = (\bar{1}_{n_i}/\sqrt{n_i})\ (\bar{1}_{n_i}^\top/\sqrt{n_i}) = \bar{\mathbf{1}}_{\mathbf{n_i}}/n_i$, where $\bar{\mathbf{1}}_{\mathbf{n_i}}$ is an $n_i \times n_i$ matrix of ones. Then

$$u(t) = -\kappa B^\top \begin{bmatrix} \frac{\bar{1}_{n_1}}{n_1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \frac{\bar{1}_{n_2}}{n_2} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \frac{\bar{1}_{n_N}}{n_N} \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_N(t) \end{bmatrix}, \tag{5}$$

where

$$y_i(t) = \bar{1}_{n_i}^\top x_i(t)$$

is the cumulative stock of buffer i, namely the sum of all the state variables of the ith subsystem (node).

In principle, in order to check whether Assumption 2 is satisfied, according to Proposition 1 we should know the matrices $A^{(i)}$. Yet this is not necessary if Assumption 3 holds, according to the next corollary.

Corollary 1: Under Assumption 3, stabilizability is equivalent to

$$\text{rank}\left(\begin{bmatrix} \bar{1}_{n_1}^\top & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \bar{1}_{n_2}^\top & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \bar{1}_{n_N}^\top \end{bmatrix} B \right) = N.$$

The proof of Corollary 1 will be given along with the proof of Theorem 1.

A. Proof of the main result

Given vector $e_n = [1/\sqrt{n}\ \dots\ 1/\sqrt{n}]^\top$, we denote by E_n its orthonormal complement:

$$\begin{bmatrix} e_n^\top \\ E_n^\top \end{bmatrix} \begin{bmatrix} e_n & E_n \end{bmatrix} = I_n.$$

Since any matrix $A^{(i)}$ has zero sum columns, we have

$$\begin{bmatrix} e_{n_i}^\top \\ E_{n_i}^\top \end{bmatrix} A^{(i)} \begin{bmatrix} e_{n_i} & E_{n_i} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ E_{n_i}^\top A^{(i)} e_{n_i} & E_{n_i}^\top A^{(i)} E_{n_i} \end{bmatrix},$$

where $E_{n_i}^\top A^{(i)} E_{n_i}$ has only negative real part eigenvalues, because 0 is a simple eigenvalue by assumption.

Proof: Consider the orthonormal transformation

$$T = \begin{bmatrix} e_{n_1} & \mathbf{0} & \dots & \mathbf{0} & E_{n_1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & e_{n_2} & \dots & \mathbf{0} & \mathbf{0} & E_{n_2} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & e_{n_N} & \mathbf{0} & \mathbf{0} & \dots & E_{n_N} \end{bmatrix}$$

and notice that T is square and $T^\top T = I_n$. Then, with easy computations we get

$$T^\top A T = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ G & F \end{bmatrix},$$

where both G and F are block–diagonal,

$$G = \text{blockdiag}\{E_{n_1}^\top A^{(1)} e_{n_1}, \ldots, E_{n_N}^\top A^{(N)} e_{n_N}\},$$
$$F = \text{blockdiag}\{E_{n_1}^\top A^{(1)} E_{n_1}, \ldots, E_{n_N}^\top A^{(N)} E_{n_N}\},$$

and F is stable. The transformed input matrix is then

$$T^\top B = \begin{bmatrix} B_0 \\ B_S \end{bmatrix},$$

where S stands for stable. In order to express control (3) accordingly, we first notice that

$$HT = \begin{bmatrix} e_{n_1} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & e_{n_2} & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & e_{n_N} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \end{bmatrix},$$

thus we get

$$\hat{u} = -\kappa B^\top H T \hat{x} = -\kappa \begin{bmatrix} B_0^\top & \mathbf{0} \end{bmatrix} \hat{x} \tag{6}$$

and the closed–loop matrix is

$$T^\top A T - \kappa T^\top B B^\top H T = \begin{bmatrix} -\kappa B_0 B_0^\top & \mathbf{0} \\ G - \kappa B_S B_0^\top & F \end{bmatrix}. \tag{7}$$

To complete the proof, we show that B_0 has rank N and thus the $N \times N$ matrix $-\kappa B_0 B_0^\top$ is negative definite. This implies stability in view of the fact that F is a stable matrix.

If we apply the Popov criterion for stabilizability, bearing in mind that 0 is the only unstable eigenvalue, stabilizability implies that

$$\text{rank}[A|B] = \left[\begin{array}{cc|c} \mathbf{0} & \mathbf{0} & B_0 \\ G & F & B_S \end{array}\right] = n,$$

which in turn implies that B_0 has rank N. ■

Note that the last claim proves Corollary 1.

IV. In the presence of flow constraints

Along the lines in [3], we can show that the proposed network–decentralized stabilizing control is effective also in the presence of flow constraints.

Suppose that control flows are subject to capacity constraints in a box, thus each component of the control must be within a lower and an upper bound:

$$u(t) \in \mathcal{U} \doteq \{u \in \mathbb{R}^m : u_i^- \leq u_i \leq u_i^+, \forall i\}.$$

The saturation function $sat(\cdot) : \mathbb{R}^m \to \mathbb{R}^m$ is componentwise defined as follows [3]:

$$u_i = sat(\phi_i) \doteq \begin{cases} u_i^- & \text{if } \phi_i < u_i^-, \\ \phi_i & \text{if } u_i^- \leq \phi_i \leq u_i^+, \\ u_i^+ & \text{if } \phi_i > u_i^+. \end{cases}$$

We consider the network–decentralized saturated control

$$u(t) = sat[-\kappa B^\top H x(t)], \tag{8}$$

with H as in (4).

Proposition 2: Under the assumptions of Theorem 1, the network–decentralized saturated control of the form (8), such that $u(t) \in \mathcal{U}$, robustly stabilizes system (1) in presence of a constant vector $d \in \mathcal{D}$, with $D\mathcal{D} \subset -\text{int}(B\mathcal{U})$.

Proof: In presence of a saturated control and a suitable disturbance vector, the closed–loop system becomes

$$\begin{bmatrix} \dot{x}_0 \\ \dot{x}_S \end{bmatrix} = \begin{bmatrix} B_0 sat(-\kappa B_0^\top x_0) \\ G x_0 + B_S sat(-\kappa B_0^\top x_0) + F x_S \end{bmatrix} + \begin{bmatrix} \hat{d}_0 \\ \hat{d}_S \end{bmatrix}.$$

Since matrix B_0 has full row rank, according to the results in [3], x_0 converges to an arbitrarily small neighborhood of the origin, $\mathcal{B}_\kappa$, for $\kappa > 0$ large enough. Then, since matrix F is stable, the overall system is stable, regardless of the Markov chain parameters. ■

V. The case of $\lambda = 0$ multiple

If we cannot assume that $\lambda = 0$ is simple for all $A^{(i)}$, in general we might not be able to find a robust network–decentralized control. We can find a control depending on the $A^{(i)}$:

$$u(t) = -\kappa B^\top W x(t), \tag{9}$$

where $\kappa > 0$ and

$$W = \text{blockdiag}\{V_{n_1} V_{n_1}^\top, \ldots, V_{n_i} V_{n_i}^\top, \ldots, V_{n_N} V_{n_N}^\top\}. \tag{10}$$

We have denoted by V_{n_i} any orthonormal basis of the left eigenspace of the 0 eigenvalue of matrix $A^{(i)}$:

$$V_{n_i}^\top A^{(i)} = 0.$$

Theorem 2: The network–decentralized control (9)–(10) stabilizes the system.

The proof is analogous to that of Theorem 1, yet now the bases V_{n_i} are functions of the parameters: the previous robust control is no more suitable. To explain this "pathology" we can consider the very simple example of a single node with a single control:

$$A = \begin{bmatrix} -(\alpha+\beta) & 0 & 0 \\ \alpha & -\gamma & \delta \\ \beta & \gamma & -\delta \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \alpha, \beta > 0.$$

This system is stabilizable iff either γ or δ are positive. If $\gamma = \delta = 0$, independently of u we have

$$\beta x_2 - \alpha x_3 = \text{constant},$$

hence the distribution of queued traffic between nodes x_2 and x_3 is an invariant (uncontrollable) variable. It is also easy to see that, if we take as output variable

$$y = \bar{1}_3^\top x = x_1 + x_2 + x_3,$$

which is the total stock at the node, then, independent of the chosen output–feedback control, stabilizability is impossible.

Indeed, 0 being a simple eigenvalue is not only a sufficient, but also a necessary condition for the control (5) to work.

Theorem 3: Under Assumption 2, control (5) is stabilizing if and only if Assumption 3 holds.

Proof: Sufficiency has already been proved. To prove necessity, consider the system with output y, which we rewrite for convenience as

$$\dot{x} = Ax + Bu, \quad y = Cx,$$

where

$$\begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_N(t) \end{bmatrix} = \begin{bmatrix} \bar{1}_{n_1}^\top & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \bar{1}_{n_2}^\top & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \bar{1}_{n_N}^\top \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_N(t) \end{bmatrix}.$$

Stabilizability implies detectability (*i.e.*, observability of all the unstable eigenvalues). From the Popov observability criterion applied to the unstable eigenvalue 0, we must have

$$\operatorname{rank} \begin{bmatrix} C \\ A \end{bmatrix} = n.$$

In view of the diagonal structure of both C and A, we can reorder the blocks and get

$$\operatorname{rank} \begin{bmatrix} C \\ A \end{bmatrix} = \operatorname{rank} \begin{bmatrix} \bar{1}_{n_1}^\top & \mathbf{0} & \dots & \mathbf{0} \\ A^{(1)} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \bar{1}_{n_2}^\top & \dots & \mathbf{0} \\ \mathbf{0} & A^{(2)} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \bar{1}_{n_N}^\top \\ \mathbf{0} & \mathbf{0} & \dots & A^{(N)} \end{bmatrix} = n.$$

A necessary (and sufficient) condition for detectability is that, for each i,

$$\operatorname{rank} \begin{bmatrix} \bar{1}_{n_i}^\top \\ A^{(i)} \end{bmatrix} = n_i. \qquad (11)$$

Assume *ab absurdo* that, for one of the subsystems, the eigenvalue 0 has multiplicity greater than 1. We remind that, in a Metzler matrix with zero sum columns, the ascent of the eigenvalue 0 is necessarily 1 (*i.e.*, the largest Jordan block associated with 0 has dimension 1), because the system is marginally stable. Therefore, there exists a right eigenvector v which is orthogonal to $\bar{1}_{n_i}^\top$ [2]. Then we have

$$\begin{bmatrix} \bar{1}_{n_i}^\top \\ A^{(i)} \end{bmatrix} v = 0,$$

in contradiction with (11). ■

VI. Example of application

In order to test the proposed control scheme, we generated random instances of large networks, as already proposed among the examples in [8]. The progress with respect to [8], in which a suitable LMI was solved, is that here we propose a *robust* solution, based on the control (5).

We considered systems of the form (1), with $D = I$, and carried out numerical experiments as follows.

[2] taken any right eigenvector $\tilde{v}$, $A^{(i)}\tilde{v} = 0$, consider its orthogonal component to $\bar{1}_{n_i}^\top$: $v = \tilde{v} - (\tilde{v}^\top e_{n_i})e_{n_i}$, where $e_{n_i} = \bar{1}_{n_i}/\sqrt{n_i}$

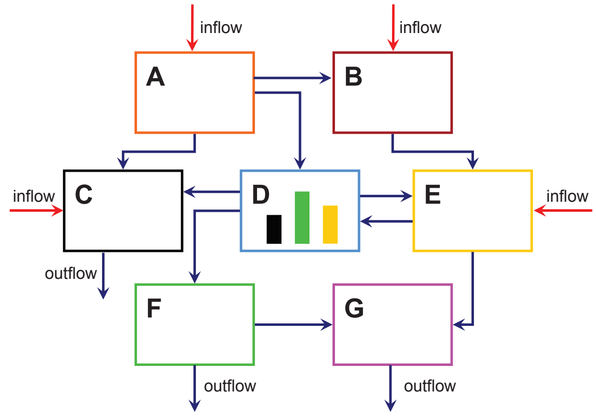

Fig. 2: A traffic problem.

1) Fix a number of nodes N and a maximum node size $\bar{n}$.
2) Randomly generate a graph in which each pair of nodes i–j is connected with probability P_c, and each node i is connected with the external environment with probability P_e.
3) For each pair of connected nodes i–j, generate a control by adding to matrix B a column k whose elements are all null except for B_{ik} and B_{jk}, which are nonnegative and nonpositive respectively.
4) Apply the proposed control to the randomly generated network and simulate the closed–loop system.

Since the matrices $A^{(i)}$ are randomly generated, we can rely on the fact that they have distinct eigenvalues, hence 0 is a simple eigenvalue. We can thus apply the control (5), which turns out to be stabilizing, as expected, whenever the system is stabilizable (*i.e.* $\operatorname{rank}[A \;\; B] = n$). In general, a random instance could be not stabilizable, due to a lack of connectivity.

These simulations are specific for a traffic congestion problem (see for example Fig. 2): traffic units, each having its own destination, enter the network at some nodes, through inflow arrows, and leave the network through outflow arrows. At each node, the traffic is logically (not necessarily physically) split in queues of elements having different directions. For instance, at node D there are three queues, composed of units directed to nodes C, E and F. Under Assumption 3, any control arc can perform its action based only on the knowledge of the total congestion at the nodes it connects. For instance, the control governs the flow along the arc connecting D to F without any knowledge of the internal splitting, yet the actual controlled flow transfers elements of node D exclusively from the queue directed to F (the green buffer inside node D, in Fig. 2).

Here we present the results of a numerical experiment. Matrices A and B were randomly generated. The external disturbance vector d, randomly generated as well, was held constant. The fixed number of nodes was $N = 10$ and the maximum node size was $\bar{n} = 5$. The randomly generated system had overall dimension $n = 42$, with $n_1 = 4$, $n_2 = 4$, $n_3 = 4$, $n_4 = 5$, $n_5 = 2$, $n_6 = 5$, $n_7 = 4$, $n_8 = 4$, $n_9 = 5$, $n_{10} = 5$. The number of controlled arcs was $m = 31$. We applied the control (5), with $\kappa = 0.8$. The simulation

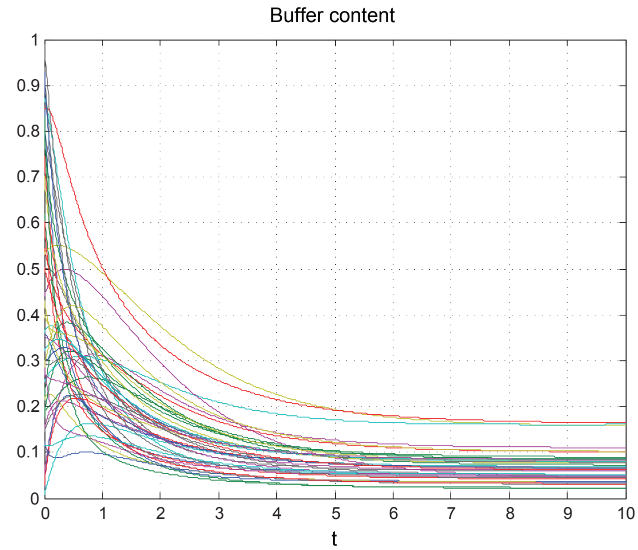

Fig. 3: Simulation results for the example in Section VI.

results, in Fig. 3, show the evolution of the controlled system starting from random initial conditions in the range $[0\ 1]$. Zero is the reference level for the buffers. Due to the presence of the persistent disturbance d, the desired set–point is not exactly reached. However, convergence to the steady state is smoothly and quite fast assured, without requiring the knowledge of the internal dynamics of the nodes.

VII. Conclusions

We have considered a traffic control problem defined on a network graph: arcs represent flow channels and nodes represent buffers, which can be partitioned into different queues associated with streams having different directions. Each node is modelled as a Markov chain. We have seen that stabilization can always be achieved with a network–decentralized control, in which the flow through each arc is governed by an agent having information only about the nodes that the arc connects.

The main result is that, when zero is a simple eigenvalue for all the Markov chains, the proposed network–decentralized strategy is robust: stability is assured independently of the specific values of the Markov chain parameters. Furthermore, we have shown that each agent simply needs information about the total amount of congestion at each of the nodes it connects. Hence, the proposed control works without requiring any knowledge of the internal traffic splitting statistics and of the actual distribution of node traffic among the different queues. The proposed network–decentralized control assures robust stabilization even when control flows are subject to capacity constraints.

Further developments of this work include the case in which constraints are present on both state and flow control variables. In particular, dealing with positivity constraints and with upper bounds due to the buffer size would be important. Another interesting question is whether, under special assumptions, we can find more specific but stronger results. For instance, if B is an incidence matrix associated with a network, we could investigate a control of the form $u = -\kappa B^T x$, considered in [3] for systems without node dynamics, and try to establish if such a control is stabilizing, or optimal in some way, for networks with traffic splitting at the nodes.

References

[1] A. Atamturk and M. Zhang, "Two–stage robust network flow and design under demand uncertainty," *Operations Research*, vol. 55, no. 4, pp. 662–673, 2007.

[2] B. Ataslar and A. Iftar, "A decentralized control approach for transportation networks," *Proc. of the 8th IFAC Symposium. on Large Scale Systems*, pp. 348–353, 1998.

[3] D. Bauso, F. Blanchini, L. Giarré, and R. Pesenti, "The linear saturated control strategy for constrained flow control is asymptotically optimal," *Automatica*, Vol. 49., no. 7, pp. 2206–2212, 2013.

[4] D. Bauso, F. Blanchini, and R. Pesenti, "Optimization of Long-run Average-Flow Cost in Networks with Time-Varying Unknown Demand," *IEEE Trans. on Autom. Control*, vol. 55, no. 1, pp. 20–31, 2010.

[5] D. Bertsimas and A. Thiele, "A Robust Optimization Approach to Inventory Theory," *Operations Research*, vol. 54, no. 1, pp. 150–168, 2006.

[6] F. Blanchini, F. Rinaldi, and W. Ukovich, "Least Inventory Control of Multi-Storage Systems with Non-Stochastic Unknown Input," *IEEE Trans. on Robotics and Automation*, vol. 13, pp. 633–645, 1997.

[7] F. Blanchini, E. Franco, G. Giordano, "Structured–LMI Conditions for Network–Decentralized Control," *52^{nd} IEEE Conf. on Decision and Control*, Florence, December 2013.

[8] F. Blanchini, E. Franco, G. Giordano, "Network-Decentralized Control Strategies for Stabilization," *IEEE Trans. on Autom. Control*, to appear.

[9] F. Blanchini, S. Miani, and W. Ukovich, "Control of production-distribution systems with unknown inputs and system failures," *IEEE Trans. on Autom. Control*, vol. 45, no. 6, pp. 1072–1081, 2000.

[10] E. K. Boukas, H. Yang, and Q. Zhang, "Minimax production planning in failure–prone manufacturing systems," *Journal of Optimization Theory and Applications*, vol. 82, no. 2, pp. 269–286, 1995.

[11] A. Ephremides and S. Verdú, "Control and Optimisation Methods in Communication Networks," *IEEE Trans. on Autom. Control*, vol. 34, pp. 930–942, 1989.

[12] A. Iftar, "A decentralized routing control strategy for semi-congested highways," *Proc. of the 13th IFAC World Congress*, Vol. P, pp. 319–324, 1996.

[13] A. Iftar, "A linear programming based decentralized routing controller for congested highways," *Automatica*, vol. 35, no. 2, pp. 279–292, 1999.

[14] A. Iftar and E. J. Davison, "Decentralized Robust Control for Dynamic Routing of Large Scale Networks," *Proc. of the American Control Conf.*, San Diego, CA, pp. 441–446, 1990.

[15] A. Iftar and E. J. Davison, "Decentralized control strategies for dynamic routing," *Optimal Control Applications and Methods*, vol. 23, pp. 329–355, 2002.

[16] R. E. Larson and W. G. Keckler, "Applications of Dynamic Programming to the Control of Water Resource Systems," *Automatica*, vol. 5, pp. 15–26, 1969.

[17] J. C. Moreno and M. Papageorgiou, "A Linear Programming Approach to Large-scale Linear Optimal Control Problems," *IEEE Trans. on Autom. Control*, vol. 40, pp. 971–977, 1995.

[18] F. H. Moss and A. Segall, "An Optimal Control Approach to Dynamic Routing in Data Communication Networks," *IEEE Trans. on Autom. Control*, pp. 1–7, 1978.

[19] S. Mudchanatongsuk, F. Ordóñez, and J. Liu, "Robust Solutions for Network Design under Transportation Cost and Demand Uncertainty," *Journal of the Operational Research Society*, vol. 59, pp. 652–662, 2008.

[20] F. Ordóñez and J. Zhao, "Robust capacity expansion of network flows," *Network*, vol. 38, pp. 136–145, 2007.

[21] H. Sarimveisa, P. Patrinos, C. D. Tarantilis, and C. T. Kiranoudis, "Dynamic modeling and control of supply chain systems: A review," *Computers & Operations Research* vol. 35, pp. 3530–3561, 2008.

[22] E. A. Silver and R. Peterson, *Decision System for Inventory Management and Production Planning*, Wiley, New York, NY, 1985.

[23] J. Wei and A. van der Schaft, "Stability of dynamical distribution networks with arbitrary flow constraints and unknown in/outflows," *52^{nd} IEEE Conf. on Dec. and Cont.*, Florence, December 2013.