Decentralized Queue Balancing and Differentiated Service Scheme Based on Cooperative Control Concept
Sabato Manfredi, Member, IEEE

Abstract—In this paper, we introduce the concept of a bottleneck-routers cooperation in the explicit rate-control framework of communication networks in order to mitigate congestion effects on the network performance and balance the queues. The proposed controller at each router (server or switch) regulates the rates of the heterogeneous source classes leveraging on the cooperation of neighboring bottlenecks. We consider the model of multibottleneck network in the presence of time delay and formulate global stability conditions suitable for network parameters and controller gains design. The proposed approach guarantees good performance in terms of link utilization, packet loss and fairness. Additionally it is guaranteed queue balancing without requiring rerouting or hop-by-hop operation differently from the existing approaches. A validation is carried out by a discrete packet experiment simulator in a realistic multibottleneck scenario to demonstrate the effectiveness of the key idea of the paper. Finally the proposed scheme is compared to some of well-known network controller-type presented in the literature in both steady-state and dynamic network scenario.

Index Terms—Cloud computing, congestion control, content delivery networks, cooperative control, industrial application, network performance evaluation.

I. INTRODUCTION

TODAY’S Internet only provides best effort service by processing traffic as quickly as possible without guaranteeing any Quality of Service (QoS)[1]. With the recent increased of demands for Internet service quality it is becoming apparent the business opportunity for the web-companies in developing several service classes will likely be demanded. The introduction of new types of services in the fixed and mobile communication networks underlines as the problem of network congestion control remains a critical issue. In the recent years several methods to control the source rate have been introduced for dealing with the drawbacks of the current congestion control methods. To support differentiated traffic, the communication forums (i.e., IETF [2], ATM [3]) have defined different service traffic classes of which some of them responds to network congestion by means of a feedback control mechanism. More specifically, it is defined a gold (or premium) service for application with stringent delay and loss requirements. Examples of applications include real-time control, audio/video transmissions. The bronze (or ordinary) service usually is not sensitive to service rates nor delays but is sensitive to packet loss so that the throughput of a connection can be decreased as much as necessary, in order to alleviate congestion. In communication networks, the bronze class is served only if there is some bandwidth left by the gold class which get the higher scheduling priority. So, at a given switch or router buffer, when both gold and bronze traffic are backlogged, the packets from the higher quality of service traffic are processed first, and the ordinary traffic is served only if there is some bandwidth left by the premium traffic. Hence, if the rate of each ordinary source traffic class is not controlled, congestion may be caused. In the rate control framework a feedback may be in the form of an explicit or implicit rate provided on an end-to-end basis via signaling (i.e., explicit rate cell, explicit congestion notification). In this way associated with each switch or link buffer there is a rate or congestion controller that computes the explicit rate for each user in order to efficiently allocate the unused bandwidth of premium traffic to the ordinary traffic and avoid buffer overflow. Typical services of ordinary traffic include event-based applications, image and data retrieval, sensor network-based surveillance and monitoring applications. Many papers in the literature under different protocols (i.e., TCP/IP, ATM) focused on the problem of designing the rate or congestion controller at bottleneck node dealing with ordinary traffic flows provided that the primary traffic is allocated sufficient capacity. The idea of using explicit feedback to perform congestion control has been explored in the wired context and more recently in wireless network environment. Seminal works on ATM network proposed mechanisms for providing explicit rate control for ordinary ABR (available bit rate) traffic. Then, in the recent years many works have been exploited Active Queue Management (AQM) schemes in TCP/ip-ATM networks to deliver preemptively congestion notification to the source for reducing its transmission rate and therefore avoiding buffer overflow. Example of design of rate controller for ATM networks can be find in [14]. Examples of AQM strategies to improve the performance of the existing protocols are proposed in [22]. Other approaches considered the problem of explicit rate control [6], [7]. In [7] the authors explore a new congestion control algorithm (e.g., Rate Control Protocol (RCP)) so that a router assigns a single rate to all flows that pass through it. In the wireless communication context, there have been several proposals (e.g., see [8] and references therein). The recent increasing diffusion of remote and web application is motivating the development of congestion


S. Manfredi is with the Department of Electrical Engineering and Information Technology, University of Napoli “Federico II,” Italy (e-mail: sabato.manfredi@unina.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2013.2265879
and rate control schemes in order to face with the requirement of the specific application (i.e., [9]–[11]). Among up to date architecture for pervasive web applications, it is worth to mention Content Delivery Network (CDN) and cloud computing that are network aimed to distribute contents to the users [12].

All the aforementioned approaches concern the design of the rate regulator that uses only local information at the switch or router for control purpose (e.g., queue length, virtual rate, link/ server capacity, incoming traffic estimation) and most of their theoretical results refer to the case of single bottleneck scenarios. In the recent years distributed coordination of multitenant systems have received significant attention (e.g., see [15] and reference therein). One common feature of this research is to allow every network agent to automatically address a common objective using only local information received from its neighboring agents. While different design techniques for cooperative control of multitenant systems have been proposed and successfully used in several applications such as formation flight, robot swarm, Lagrangian systems, sensor networks ([16], [18]–[20]), renewable energy [17], distributed fault tolerant control ([21]), the benefits of its application to communication network is quite unexplored. In this direction, recently in [4] it is proposed a controller scheme for DiffServ Network that is partially based on the idea of formation control [5]. Specifically it is used a leader-follower scheme on a link capacity between premium buffer and ordinary buffer at each server. In this paper we extensively apply the multi agent approach to communication network by introducing for the first time a concept of a cooperation-based rate control in the explicit-rate control framework of communication networks. Differently all the exiting approaches in the literature, herein each router (server or switch) regulates the rates of the heterogeneous sources explicitly leveraging on the cooperation of neighboring bottlenecks. To our knowledge the development of rate control scheme based on cooperative theoretic concept is quite unexplored as well as its application to realistic network communication scenarios. We firstly present a multibottleneck model and introduce the concept of "overlay virtual graph" that easily allows to recast the congestion problem in terms of cooperative control. Then, we will present the Cooperation-based Rate Control scheme (in the follows briefly CRC) in order to: 1) stabilize the network and give sufficient stability conditions suitable for network parameters and controller gains design; 2) guarantee max-min fairly bandwidth allocation to the heterogeneous sources; 3) balance the network queue length at a desiderate set point value reducing packet loss and improving link utilization. Differently from the standard existing approaches in the literature on the queue balancing (i.e., [12]), herein the server load balancing and end-to-end performance are guaranteed without requiring rerouting or hop-by-hop operation. The approach can be easily extended to differentiated service (DiffServ) network and applied to the different network technology or web-bases application scenarios (Internet, content delivery network, cloud computing) and therefore it is of interest for the industrial community. Moreover, the presence of multibottleneck and heterogeneous sources time delay is taken into account in the problem formulation. Finally, we use packet experiment simulator to demonstrate that the proposed control can be implemented and that it achieves the network desired behavior in a more realistic multibottleneck scenarios. The rest of the paper is outlined as follows. In Section II, multibottleneck model and the definition of virtually bottleneck neighbors and overlay graph used in this paper are defined. In Section III, a cooperation based rate control and the closed loop stability conditions are presented and in Section III-A the performance issues are outlined. The effectiveness of the control law is validated, tested, presented, and compared than existing well-known network controller-type in the literature through packet numerical simulations in Section IV. Finally, conclusions, ongoing work, and extension to other application of the proposed approach are outlined in Section V.

Notation

Given $z \in \mathbb{C}$, $|z|$ denotes its absolute value. For a set $V \subset \mathbb{C}$, $|V|$ denotes the cardinality of $V$. For a square matrix $B$ with real eigenvalues, $\lambda(B)$ denotes its spectrum, $\lambda_m(\lambda_M)$ denotes the algebraically smallest (largest) eigenvalue. $s$ denotes a Laplace complex variable. Let $G(N, E, A)$ be a graph with the set of nodes $N$, set of edges $E \subseteq N \times N$, and an adjacency matrix $A = \{a_{ij}\}$ with nonnegative adjacency elements, the set of neighbors of the $i$-th node is defined by $N_i = \{k \in N : a_{ik} = 1\}$. Considering an indirect graph, the degree value $d_i$ of the node $i$-th is the number of the neighbors of the node $i$-th. The Laplacian is a matrix $L$ of elements $l_{ij}$ such that $l_{ij} = \sum_{j=1}^{N} a_{ij}$ if $i = j$, $l_{ij} = -a_{ij}$, if $i \neq j$. The Laplacian can be recast as $L = D - A$ with $D$ is the $N \times N$ diagonal matrix having in position $i$-th the degree value $d_i$ of the node $i$-th. We define the extended Laplacian matrix of elements $\tilde{l}_{ij}$ as $\tilde{L} = L + I$, with $I$ being the identity matrix of opportune dimensions. The matrix $\tilde{L}$ associated to the undirected graph is real and symmetric with real spectrum $\lambda(\tilde{L})$.

II. NETWORK MODEL AND OVERLAY VIRTUAL GRAPH

In the recent years various dynamic models have been used by a number of researchers to model a wide range of queueing and contention systems. Several variants of the fluid model have been extensively used for network performance evaluation and control (e.g., [14], [22]). Herein, the main objective is to consider a low order complexity model of multibottleneck capturing the essential dynamics of network behavior which is suitable for a distributed cooperative control design. Moreover, we would consider in the model the presence of time-delays in the sources data-flow. A time-delay is due the time elapsed between a rate command signal by a switch controller and the actual time this rate is set. This delay from the control input to the regulated output is the sum of two delays (backward delay $\tau_i$ from controller to source and forward delay $\tau_j$ from source to controller) named the Round-Trip Time delay RTT. Considered a network graph consisting by a set of congested links $N = \{1, 2, \ldots, n\}$ and $M = \{1, 2, \ldots, m\}$ accessing source classes by a specific source-destination path, the source-link interconnections can be described by the routing-matrix

$$R_{ij}(s) = \begin{cases} e^{-s\tau_i}, & \text{if source } j \text{ traverses link } i, \\ 0, & \text{otherwise} \end{cases}$$

with $\tau_{i,j}$ denoting the delay of the source $j$ with respect to (w.r.t) link $i$. In what follows, for sake of brevity, we use the term
source to briefly denote source class, namely, the set of sources characterized by the same service class or priority. We introduce the forward routing matrix $R(s)$ of elements $e^{-\tau_{f,ij}}$ with $\tau_{f,ij}$ is the forward time delay from source $j$ to link $i$, and the backward routing matrix $R_b(s)$ of elements $e^{-\tau_{b,ij}}$ with $\tau_{b,ij}$ is the backward time delay from link $i$ to source $j$. In this way, the source $j$-th has w.r.t link $i$-th the round trip time: $RTT_{i,j} = \tau_{f,ij} + \tau_{b,ij}$. Starting from the simple fluid queue model of a single bottleneck and multiple time delayed sources widely used in the literature (i.e., [14], [22]), and denoted $q(t)$; the queue length at the bottleneck link $i$-th and $r_i(t)$ being the non negative data flow rate of the $j$-th source accessing to the $i$-th bottleneck link, the network open loop dynamic model is described by

$$q_i(t) = \sum_{j} r_{i,j}(t) - c_i(t)$$

(1)

for $i \in N, j \in S_i = \{s \in M : s$ across the link $i$-th$\}$ and $c_i(t)$ being the rate at which data is sent out from the link $i$-th. In order to recast the congestion problem in terms of cooperation control concept, we introduce the set of virtually bottleneck neighbors of $i$-th link defined as $N_i = \{k \in N : S_i \cap S_k \neq \emptyset, a_{ik} = 1\}$. In other words, virtually bottleneck neighbors are bottlenecks sharing source paths. For instance referring to Fig. 1, links $1 \leftrightarrow 2$ and $2 \leftrightarrow 6$ share the path of the source $S_1$ and so they are virtually bottleneck neighbors. In the same way the links $4 \leftrightarrow 5, 5 \leftrightarrow 2$ and $2 \leftrightarrow 6$ are one step virtually neighbors. On the other side, links $1 \leftrightarrow 4$ and $4 \leftrightarrow 3$, although they are physical connected, are not virtually bottleneck neighbors. The overall graph composed of bottleneck nodes and their virtual neighbors is defined as the overlay virtual graph (denoted by highlighted solid lines in Fig. 1). According to the rate control strategies presented in the literature (see, i.e., [6] and references therein), we consider that source rate $r_{i,j}(t)$ will be assigned to the source $j$-th by a feedback controller $u_{i,j}$ located at the bottleneck $i$-th. Therefore starting from the open loop model (1), we get the following closed loop model:

$$q_i(t) = \sum_{j} u_{i,j}(t) - RTT_{i,j} - c_i(t)$$

(2)

for $i = 1, \ldots, n$ and with $RTT_{i,j}$ is the round trip time of the source $j$-th w.r.t link $i$-th. We note that the source rate commands $u_{i,j}$ should satisfy the constraint on the aggregate available rate $u_i$ computed by the controller. So if $u_{i,j} = k_{i,j}u_i$, $k_{i,j}$ are non negative controller gains to be designed so that $\sum_{j \in S_i} k_{i,j} \leq 1$. Let us assume that the final allocated rate $r_j$ to the source $j$-th, $\forall j \in M$, is equal to the minimum rate value $u_{m,j}$ among the rate values assigned by the links along the path of its flow (i.e., $r_j = u_{m,j} = \min_{i \in B_j} \{t \in N : l$ is a bottleneck for the source $j$}). Because the minimum operation is taken over a finite number of links and each flow $j$ has at least one bottleneck on its path, there should exist $u_{m,j}, \forall j \in M$. Therefore, the vector $r = [r_1, \ldots, r_m]^T$ denotes the allocated sources rates.

We do the following assumptions:

**Assumption 1**: The sources are persistent until the closed-loop system reaches steady state meaning that the source always has enough data to transmit at the allocated rate.

**Assumption 2**: All links are bottleneck for at least one source $j \in M$. Specifically the link $i$-th is a bottleneck for a given source $j$-th if and only if the source $j$-th has the maximum rate among all sources using the link $i$-th (i.e., $r_j \geq r_{j'}$, for all $j' \in S_i$). In other words, the bottleneck $i$-th is a link which is limiting for a given allocation $r_j$. Moreover, being all links bottleneck for at least one source we can assume $c_i(t) = c_i$ for all $i$, with $c_i$ to be the $i$-th link capacity.

Notice that in the above arguments we considered constant time delays $RTT_{i,j}$ as well as in several approaches in the literature as in [4], [14], [22]. The time delay includes propagation, queuing, processing and transmission components. From the theoretical point of view this means we are dealing with high speed network for which $q(t)/c_i \ll 1$ for all $i$. In this way the time varying queuing delay is neglected and the overall time delay can be well approximated as constant value.

### III. Cooperation Based Rate Control Scheme

Herein we recall and then extend the basic features of the Cooperation based Rate Control scheme (shortly CRC) recently introduced in [24]. The controller is implemented at the bottleneck node and adjusts sources rate according to both its own congestion level (i.e., queue length) and that of its virtually bottleneck neighbors. Specifically, in [24], the following sufficient condition was given for network closed loop stability and queue balancing that can be used for controller gains and network parameters design.

**Theorem 1 ([24]):** Consider a n-links m-sources communication network described by (2). Chosen the cooperative control action

$$u_{i,j}(t) = k_{i,j} \sum_{k \in N_i \cup \{i\}} (q_k(t) - q_i(t)) + k_{f,i} \dot{c}_i(t)$$

(3)

then the following hold:

a) the network is globally asymptotically stable if

$$k_{i,j} < \frac{\pi}{2|S_i|RTTM_i,\lambda_M}$$

(4)

$\forall i \in N, \forall j \in S_i$, with $RTTM_i = \max\{RTT_{i,j}, j \in S_i\}, \lambda_M$ maximum eigenvalue of $L, k_{f,i}$, $\dot{c}_i(t)$ is a feed-forward action for link capacity allocation with gain $k_{f,i}$ and $\dot{c}_i$ is the estimation of the link capacity.
b) the network queues asymptotically converge to the same set point value $q_0$ with resulting queue balancing state.

The control law $u_{i,j}(t)$ is composed of the feedback cooperative term $u_{i,j}^{fb}(t) = k_{f,i} \sum_{k \in \mathcal{N}_i} (q_k(t) - q_i(t))$ (including the set point term in $q_0$) and of the feedforward action term $u_{i,j}^{ff}(t) = k_{f,i} c_i(t)$. Hence, we have to design feedback gains $k_{f,i}$ and feedforward gains $k_{f,i}$. The tuning of the feedback gains $k_{f,i}$ can be carried out by using the stability condition (4) based on the network parameters. From Assumption 2 it results $c_i(t) = c_i$ and thus $u_{i,j}^{ff}(t) = k_{f,i} c_i(t) = k_{f,i} c_i$. In [24] the gains $k_{f,i}$ was designed according to

$$u_j = \sum_{k \in \mathcal{N}_i} w_{ik}$$

(5)

with $w_{ij}$ being the priority-weight associated to the source class $j$-th. The amount of capacity allocated to the $j$-th source then results: $u_{i,j}(t) = w_{ij} / \sum_{k \in \mathcal{N}_i} w_{ik}$. In this way, the allocation of the available capacity among sources guarantees not only that the allocated capacity is within bound $c_i$ but also that the allocation is proportionally fair. With proportional fairness, sources with greater weights $w_{ij}$ are allocated a larger amount of capacity. We can interpret $w_{ij}$ as a preassigned level of Quality of Service within the ordinary class to the source $j$-th. Thus (5) can be used for feedforward gains $k_{f,i}$. Design purpose in order to fairly allocate the available capacity $c_i$ on the base of source priorities or differentiated service requirements. This easily allows to apply the approach to DiffServ network. In the case of $w_{ij} = w_i$ for all $j \neq i$, the resulting capacity allocation is max-min fair with all sources getting the same resource quota. We refer the reader to [24] for more details about the prove of the Theorem 1. The proposed control scheme guarantees queue balancing and set point regulation by opportunely tuning the feedback and feedforward gains (i.e., $k_{f,i} = w_{ij} / \sum_{k \in \mathcal{N}_i} w_{ik}$ and $k_{f,i} < \pi / 2|S_i|RTTM_i \lambda M$) on the base of network and source features (i.e., virtual graph topology, source priority, maximum round trip time). In [24] it is reported the above Theorem 1 and a comparative performance evaluation of the proposed CRC algorithm at the higher layer level than the transport one. This in order to assess that the proposed strategy could be implemented according to any protocol or technologies by using the available field of the specific control packet to send queue information among the neighboring links: for example 1) over ATM protocol, the RM cell can be used; 2) over TCP protocol, it is possible to use the same packet signaling proposed in [7]; 3) over wireless protocol it is possible to use the "HELLO" packet, that is a special packet periodically sent from a node simultaneously to other routers to discover neighboring routers; 4) over CDN network, a control signal at the application layer can be used. As stated above, the feedback design condition (4) depends on the round trip time and network topology information that can be not easily available by the designer. In the following, we first introduce Corollaries of Theorem 1 in order to give practical control design law depending on more accessible network parameters. This is particularly appealing for the network industrial applications.

Corollary 1: Consider a n-links m-sources communication network described by (2). Let $\tau_{M,i}$ be the propagation delay of the connection between the source $j$-th and the link $i$-th and chosen the cooperative action control (3), then the network is globally asymptotically stable if

$$k_{f,i} < \frac{\pi}{2 S_i \left( 2 \tau_{M,i} + \frac{b_i}{c_i} \right) \left( 2 S_i + 1 \right)}$$

(6)

$\forall i \in \mathcal{N}, \forall j \in \mathcal{S}_i$, where $b_i$ is the buffer size, $\tau_{M,i} = \max \{ \tau_{M,i,j}, j \in \mathcal{S}_i \}$, $S_i = \max \{ S_i \}$. Moreover, the network queues asymptotically converge to the same set point value $q_0$ with resulting queue balancing state.

Proof: Let $b_i$ the buffer size of the $i$-th link, thus $RTTM_i = 2\tau_{M,i} + b_i/c_i$, with $\tau_{M,i}$ is the maximum propagation delay among the sources accessing to the $i$-th link. Moreover, from the Gersgorn’s theorem all the eigenvalues of $L$ are located in the union of the following n disks: $l_i - \{ z \in \mathbb{C}, z - l_i, < S_{ij} = \{ z \in \mathbb{C}, |z - d_i| \leq \sum_{j=1, j \neq i}^{N} |a_{ij} |, i = 1, \ldots, N \}$, where $d_i = \max |a_{ij}|$, $i = 1, \ldots, N$, $d_M$ be the maximum degree of the virtual graph, and defined the largest disk radius $r_M = \max \{ \sum_{j=1, j \neq i}^{N} |a_{ij} |, i = 1, \ldots, N \}$, being $L$ real and symmetric matrix then follows: $\forall \lambda \in \mathbb{C}, 1 \leq \lambda \leq 2d_M + 1$. Let $S_M = \max_i S_i$ and being $d_M = \max d_i$ and hence we can recast (4) into the relation (6). The proof of the second part of the Corollary follows the same arguments given in the Theorem 1.

Corollary 2: Consider a n-links m-sources communication network described by (3). Let $\tau_{M,i}$ be the propagation delay of the connection between the source $j$-th and the link $i$-th and chosen
the cooperative control action (3), then the network is globally asymptotically stable if
\[
k_{i,j} \leq \frac{\pi}{2 |S_i| \left(2 \tau_{p,i} + \tau_j \right) S_M}
\]
\(\forall i \in N, \forall j \in S_i\), where \(k_i\) is the buffer size, \(\tau_{p,i} = \max\{\gamma_{i,j}, \tau_j \} \), \(S_M = \max\{|S_i| + |S_j| : (i,j) \in N \times N\}\). Moreover, the network queues asymptotically converge to the same set point value \(q_0\) with resulting queue balancing state.

**Proposition 1:** Let \(\lambda_M\) the maximum eigenvalue of the extended Laplacian \(\tilde{L}\), if the virtual network graph associated to \(\tilde{L}\) is connected and bipartite then results (see i.e., [25] and references there in) and how those results can be used for distributed network design because they relate network parameters (e.g., link capacity \(c_i\), buffer size \(b_i\), propagation delay \(\tau_{p,i}\), and number of source classes) to feedback controller gains \(k_{i,j}\). Notice that usually the number of sources is much larger than the number of links (\(m \gg n\)). This implies that the virtual graph is undirected and connected or at least is composed of the number of connected and undirected clusters. Also the assumption of bipartite graph is a realistic assumption for a virtual graph associated to the communication network. Indeed, it has shown as communication physical and overlay network topologies have scale free property (see i.e., [25] and references there in) and how those networks can be reviewed as bipartite graph [26].

### A. Performance Issues

Herein, we analyze the CRC performance in terms of link utilization, set point regulation and fairness of the proposed controller.

1) **Set Point Regulation and Queue Balancing:** From the proof of the part b) of Theorem 1 we have shown that the CRC algorithm assures queues set point regulation to a desired \(0 < q_0 < \min b_i\) with resulting network queue balancing and set point regulation. Moreover this avoids packet dropping for buffer overflow.

2) **Link Utilization:** From Theorem 1 and Corollaries 1 and 2, if the cooperative control law (3) is applied, at the steady state results: \(\sum_{j \in S_i} r_j = c_i\) implying that the capacity at the link \(i\)-th is fully utilized.

3) **Fairness:** In a shared environment the throughput for a source depends upon the demands by other sources. The most common criterion for the correct share of bandwidth for sources in network environment is the so called max-min allocation [32]. It provides the maximum possible bandwidth to the source receiving the least among all contending sources. Notice that max-min allocation is both fair and efficient in the sense that all sources get an equal share on every link and that each link is utilized to the maximum load possible. In the following, we will show how CRC achieves max–min fair resource allocation.

**Definition 1:** A vector of allocated rates \(r = [r_1, \ldots, r_M]^T\) is feasible if \(r_j \geq 0, \forall j \in M\) and \(\sum_{j \in S_i} r_j \leq c_i\), for all \(i = 1, \ldots, n\).

**Definition 2 ([31]):** A vector \(r\) is max-min fair if and only if it is feasible, and for each \(j \in M\) and for any other feasible vector allocation \(r^\prime\) for which \(r_j < r_j^\prime\), there is some \(j^\prime\) such that \(r_j^\prime \geq r_j^\prime\).

Max-min fair rate vector is such that for every rate \(r_j\), any attempt to increase \(r_j\) must result in a decrease of another rate \(r_j^\prime\), for which \(r_j > r_j^\prime\) in order to maintain feasibility. In this way it is given priority to flows with small rate values.

**Proposition 1:** Let the network (2), then the vector \(r\) of the source allocated rates by the cooperative rate control (3) is max-min fair.

**Proof:** We introduce the set of flows bottlenecked at the link \(i - th\), \(S_i = \{ j \in M : r_j = u_{i,j} = u_{i,j}^0\}\), and the set of all flows not bottlenecked at link \(i - th\) and traversing it \(S_i^{unb} = \{ j \in M : r_j = u_{i,j}^0 = w_{i,j}^0\}\). The above Corollaries can be used for distributed network design because they relate network parameters (e.g., link capacity \(c_i\), buffer size \(b_i\), propagation delay \(\tau_{p,i}\), and number of source classes) to feedback controller gains \(k_{i,j}\). Notice that usually the number of sources is much larger than the number of links (\(m \gg n\)). This implies that the virtual graph is undirected and connected or at least is composed of the number of connected and undirected clusters. Also the assumption of bipartite graph is a realistic assumption for a virtual graph associated to the communication network. Indeed, it has shown as communication physical and overlay network topologies have scale free property (see i.e., [25] and references there in) and how those networks can be reviewed as bipartite graph [26].

### IV. CONTROLLER VALIDATION

Now, we shall seek to validate the effectiveness of the CRC controller derived above in a realistic multibottleneck scenarios and compare its performance with respect to the well known different type of controller schemes in the literature such as PID-based ([22], [28]), Smith predictor-based [14] and standard rate-based ([29], [30]). For sake of brevity, we refer the reader to the above references for more details. To this aim we used the NISTHFC network simulator [27] (in the follows shortly NIST), a packet network experiment simulator developed to provide a means for researchers and network planners to analyze the behavior of networks and to implement explicit rate control strategy. We have been built a library extending the NIST simulator functionality. New application components (which are in charge of data treatment) and new agent has been
added to allow the simulation of data transferring. The library has been designed to extract switch queue status information and to send such data to the switchs neighbors. Simulation experiments refer to the general multibottleneck topology composed of 50 switches connected by links with capacity value falling in the range [155–170] Mb/s. Ordinary sources have minimum bit rate of 100 Mb/s and maximum bit rate of 160 Mb/s. The target queue length \( q_{th} \) is set to 60% (e.g., 0.60) of the buffer size \( b_i \) = 300 cells, \( \forall i \). The CRC feedforward and feedback gains are tuned for max-min resource allocation purpose such that all sources receive the same allocation quota. Specifically, source classes have assigned the same priority \( w_j \) and \( k_{i,j} = 1/ \bar{S}_i, \forall j \in \bar{S}_i \), while the CRC feedback gains are designed according to (7) and such that \( k_{j,j} = k_i, \forall (i,j) \in N \times \bar{S}_i \). The sampling period is 2 ms. Namely, we investigate the effectiveness of the CRC algorithm in balancing the network queue length at a desired set point in a multibottleneck scenario and heterogeneous sources. Moreover we evaluate the controller performance computing link utilization and JAIN index [32]. By and large, the latter index quantifies the fairness of \( m \) sources and ranges from 1/\( m \) (worst case) to 1 (best case): in our case, it is maximum when the allocation is max-min fair (i.e., all source classes receive the same allocation because they have the same priority). The control parameters of the other controller schemes were selected for this network scenario in accordance to the guidelines given in the original papers [14], [22], [28]–[30].

### A. Queue Length Stabilization and Balancing

First of all we would show how the CRC algorithm stabilizes the network and balances the queues length at a desiderate set point value, reducing packet loss and improving link utilization. The Table I shows the average network queue length and the standard deviation of the proposed method compared to the other schemes under increasing load conditions (e.g., number of sources \( N \in \{10, 20, 40, 80\} \)). Notice that the CRC algorithm outperforms the other scheme and guarantees no packet loss, reduced queue standard deviation and set-point regulation with a resulting queue balancing performance (even for increasing load). In addition, the queues do not present oscillation with a jitter delay variation reduction and consequently improving of the QoS perceived by the users in terms of service latency variation. Notable among the other controllers, it is the good queue set point regulation presented by the PID-based network controller, although it presents degraded performance in terms of link utilization and fairness for increasing load \( N \).

### B. Link Utilization and Fairness

For each value of the load, we computed the steady state average network link utilization. As it appears from Table I, CRC scheme achieves a good link utilization. Moreover, we evaluated the max-min fairness level of the CRC by computing the JAIN index for each couple sources/destination. The results presented in Table I clearly show the good max-min fair allocation level provided by the proposed controller even under large difference between the source-switch paths length. To further confirm the effectiveness of the proposed scheme, we computed the fairness under dynamic load variation conditions. Specifically we have varied dynamically the load and computed the allocated bandwidth at each of the source accessing an Overloaded Switch \( OS \) with link capacity \( c_{OS} = 155 \) Mb/s. In Fig. 2 the load changes dynamically in the time according the value reported on the horizontal axis while on the vertical axis is reported the allocate bandwidth to one of the source. CRC, differently from the other controllers, assures max-min fairness (e.g., the value in figure is closed to the max-min quota \( r = c_{OS}/N = 155/N \) for each value of \( N \) and in the presence of large difference between the source-switch paths length (i.e., source variation with different propagation delays). Moreover the allocated bandwidth does not present oscillation than the other strategies reducing the jitter delay variation.

### C. Scalability

Finally, from the pervious results shown in Table I and Fig. 2 it results that CRC presents a good scalability feature avoiding the performance degradation even for increasing source demands considerably exceeding the link capacities both in static and dynamic scenario.
Summing up, CRC outperforms the other controllers in terms of network average performance (Table I). This can be explained because of the lack of robustness of the other schemes to load variations, cross traffic as well as round trip variation. Additionally they present an intrinsic performance limitation in dealing with the multibottleneck scenario (drawback encompassed by the CRC algorithm by the cooperative-based control action). Moreover, the fairness (Table I and Fig. 2) is strongly degraded under the other controller schemes because they are unable to guarantee fair allocation under dynamic source variation with each one perceiving a different round trip time (due to the different source-destination path length). Notice that the proposed cooperative based approach can be applied to different network technologies or application scenarios to avoid congestion and to balance the queues while it is satisfied fairness and scalability property. For example in CDN application the router is also a destination node for some packets associated to the source requests. In this case the proposed approach can guarantee load balancing and scalability with desired end-to-end performance in terms of packet loss and fairness. In a similar way, the approach can be used to deal with the resource load balancing requirement in a cloud computing scenario.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have introduced the concept of a bottleneck switches cooperation in the explicit-rate control framework of networks to mitigate congestion effects on the network performance. We proposed a routers cooperation based rate control scheme that stabilizes and balances the network queues, reduces packet loss, improves link utilization and guarantees max-min fair allocation. Alternative and more practical controller gains design rules related to network parameters have been introduced by Corollaries 1 and 2. The controller implementation issue has been assessed by using a packet network experiment simulator. A comparison than the main controller schemes present in the literature has been addressed confirming the effectiveness of the proposed approach in multibottleneck and load dynamic scenario. The proposed Cooperation based Rate Control scheme represents an early use of router cooperation concept in the network framework and can be employed for real-time application/communication network contexts for which the rate control, the fairness and the resource balancing are focal issues such as MPLS, cloud computing, traffic shaping, wireless multimedia sensor networks, content delivery networks. Our future work will be devoted to the actual implementation of our solution in a real system.

REFERENCES


Sabato Manfredi (M’13) received the Ph.D. degree in automatic control from the University of Naples Federico II, Naples, Italy, in 2005. Since 2005, is an Assistant Professor at the University of Napoli Federico II, Napoli, Italy. His research interests fall in the field of automatic control systems, with special regard to the design and implementation of novel modeling techniques, control and identification of communication networks, multiagent systems, complex networks, and underwater breathing systems. Areas of application include networked monitoring and control system, wireless sensor networks, congestion control, and wireless networked multirobot coordination. He works on consensus algorithms, distributed control, and energy harvesting systems for wireless networks.