Using Congestion Graphs to Analyze the Stability of Network Congestion Control

Damian Hobson-Garcia and Tomohisa Hayakawa

***Abstract*—In this paper we apply the analysis of delay differential systems to congestion control of computer networks. We represent the congestion of the network as a set of graphs and use that structure to define a matrix equation that represents the queue dynamics of a computer network. We first apply the technique to analytical analysis of first-order delay differential dynamics and then show how the method can be extended to numerical analysis of higher-order dynamics in both single-link and multi-link configurations. The network model and control are based on a continuous-time fluid flow model of network traffic data rates.**

1. Introduction

OVER the past several years, the Internet has seen significant growth both in the number of users and in the amount and rate of traffic that it supports. A key component to this success has been the ability of the network to properly share available bandwidth and handle congestion. The congestion control algorithms, described by Jacobson in [1] and implemented by the current Transmission Control Protocol (TCP), have been crucial to maintaining the stability of the Internet over the past decades. That being said, a significant amount of work has been conducted in recent years to address the limitations of these algorithms and to bring a more control theoretic approach to its analysis. Several new TCP algorithms, including TCP Vegas, XCP [2] and FAST TCP [3] as well as several implementations of Adaptive Queue Management (AQM), such as the work of Hayes *e*t al. [4] and Jing *e*t al. [5] have been proposed. Many of the most promising possibilities have been adapted from some of the ideas presented by Johari and Tan [6], who define a method to control network congestion and completely characterize its stability under the presence of uniform round-trip delay across all users. The method uses a distributed end-to-end congestion control law based on a fluid flow model of Internet congestion. This analysis is further extended by Massoulié [7] to include systems with heterogeneous round-trip delays.

These analyses are based on the concept of user utility and resource pricing. Under this model, each user attempts to maximize their own utility given the current network congestion conditions. We can then think of the bandwidth allocation problem as a global optimization problem, and an attempt is made to find the most efficient method to distribute the network bandwidth among all users. The optimization problem is typically decoupled by considering the dual to the original problem, using Laplacian gradient multipliers as described by Low and Lapsley [8].

D. Hobson-Garcia and T. Hayakawa are with the Department of Mechanical and Environmental Informatics Tokyo Institute of Technology, Tokyo 152-8552, JAPAN

In this paper, we begin by analyzing the control law presented by Paganini *et al*. [9] using a time domain, Lyapunov-Krasovskii based approach, as opposed to the frequency domain analysis performed previously. In particular, we perform our analysis using matrix equations, as opposed to the sums of sets used by Papachristodoulou [10]. We define a set of congestion matrices that describe the interconnection and interdependence of the data sources and congested links in the the network. The matrix representation we propose is easily extensible to the analysis of different control laws and equilibrium conditions using numerical analysis, which we show in subsequent sections.

In Section 2 we define the specifics of the network configuration, congestion reporting methods and queue dynamics assumed for this analysis. In Section 3 we introduce the requirements and rationale behind congestion control and in Section 4 we introduce the notion of dual congestion control as it applies to this analysis. We analytically determine the stability of the first-order control law and define a criteria for stability of the system using congestion matrices in Section 5 and extend the results to a second-order system in Section 6. Examples of the single-link and multi-link configurations are given in Section 7 and Section 8 respectively. We conclude this paper in Section 9.

2. Network systems and data flow

We consider the system consisting of a network of nodes, denoted by $\mathcal{N} \triangleq \{1, \ldots, w\}$. A subset of these, $\bar{\mathcal{N}} \subseteq \mathcal{N}$, send data to some other node in $\mathcal{N}$ via the network. We call the nodes contained in $\bar{\mathcal{N}}$ "senders", and we denote the connection between any two directly connected nodes in the network as a "link". Each sender contains one or more independent "sources" of data, which can represent different tasks or programs operating at each sender. The total set of sources in the network is denoted by $\mathcal{S}_m = \{1, \ldots, m\}$. We assume that each node in the network is connected to every other node through a series of links, but is not *directly* connected. As data travels from the sender to the eventual receiver, it traverses a series of links, the collection of which we call a "path", through the network and we refer to a section of this path as a "subpath".

Each link in the network has a maximum data rate that describes the maximum speed at which data can traverse the link. If multiple sources send data on the same subpath, the links within that subpath may become congested. That is, at these links there exists a time at which an attempt is made to send data in excess of the maximum data rate for the link. In this case, a queue associated with each link begins to fill with the excess data and will eventually overflow unless the data rate is reduced. We denote the set of congested links by $\mathcal{L}_n = \{1, \ldots, n\}$, and note that this set is typically, but not necessarily, much smaller than the set of all of the links in the network. For the purposes of this analysis, we ignore all non-congested links, since their internal queue lengths are always zero and they can always transmit all data that enters them.

The data from source i is routed through a subset of the congested links, denoted by $\mathcal{L}(i) \triangleq \{x \in \{1, \ldots, n\} : i \text{ sends data on link } x\} \subseteq \mathcal{L}_n$, where $i \in \{1, \ldots, m\}$. Likewise, link j carries data that has been received from a subset of the sources, denoted by $\mathcal{S}(j) \triangleq \{x \in \{1, \ldots, m\} : j \text{ transmits data from sender } x\} \subseteq \mathcal{S}_m$, where $j \in \{1, \ldots, n\}$. Note that $\mathcal{L}(i)$ and $\mathcal{S}(j)$ are reciprocal sets in that $a \in \mathcal{L}(b)$ if and only if $b \in \mathcal{S}(a)$.

In this analysis, we assume that each of the m sources *always* have data to send on the network. We do not, however, assume that all sources begin sending data at the same time. We further assume that once the path that the data takes through the network has been established, that path will never change and thus there exists a 1-to-1 mapping between a source and the path that data from that source traverses through the network.

3. Congestion Control

The purpose of a congestion control algorithm is to determine a way to co-operatively share common network links between multiple sources so that data loss and congestion in the network are minimized. In other words, if a link capacity is exceeded, the link queue should not also be exceeded. A link is defined as common between two sources if that link carries data from both sources. The further challenge comes from the fact that the congestion control algorithm should be distributively and independently implemented at each source. Furthermore, each source should be able to set its own rate without any knowledge of the number of sources sharing a link or the sending data rate of the other sources, while still ensuring that the link capacity is not exceeded.

In order to design a congestion control algorithm, it is necessary to have a method to feedback congestion information from the links to the data senders (either through implicit or explicit means). Typically there is a time delay between generating data at the source and receiving the corresponding congestion information from the links. For our purposes we assume that this is a constant value per source and note that data must traverse the entire path from a data sender to its eventual receiver before congestion information can be generated and subsequently communicated to the data source. Thus, regardless of where along the path congestion occurs, the round-trip delay remains the same. In practice the round-trip delay is a function of the lengths of the queues in the network. In this paper, however, we consider the local stability about an equilibrium set of queue lengths. Since the round trip delay is a function of the queue lengths in the network and the queue lengths in our analysis do not fluctuate significantly from their equilibrium values, we infer that the round trip delays will likewise be relatively constant, making our simplification reasonable. Specifically, let the constant round-trip time of source i be denoted by τ_i. This round-trip delay can be divided into a forward delay (from source to link) and a backward delay (from link to source), represented by τ_i^{f} and τ_i^{b} respectively, such that $\tau_i = \tau_i^{\mathrm{f}} + \tau_i^{\mathrm{b}}$.

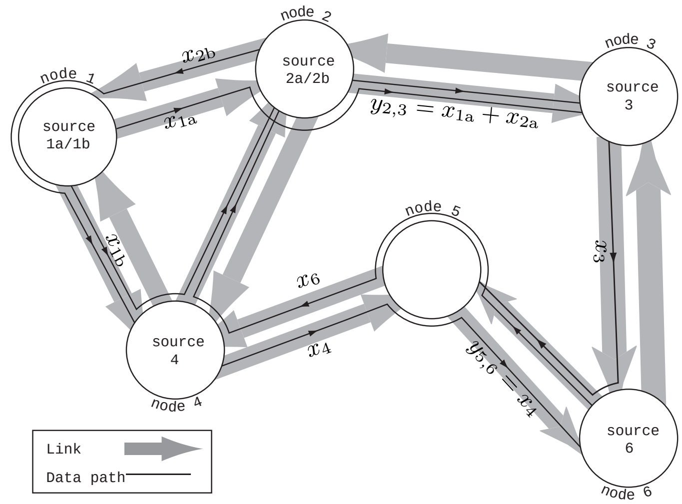

Fig. 1. Example of data flow in the network with 7 data sources

Given the above assumptions and properties, we can represent the flow of data through a sample network as shown in Fig. 1, where the set of network nodes $\mathcal{N}$ is the set $\{1, \ldots, 6\}$, while the set of senders $\bar{\mathcal{N}} \subset \mathcal{N}$ is the set $\{1, 2, 3, 4, 6\}$. In Fig. 1, node 1 has two sources, $1\mathrm{a}$ and $1\mathrm{b}$, and the same is true of node 2. The set of all sources is given by $\mathcal{S}_7 = \{1\mathrm{a}, 1\mathrm{b}, 2\mathrm{a}, 2\mathrm{b}, 3, 4, 6\}$. Each link is a unidirectional connection between two adjacent, connected nodes. As such, if we denote $[a, b]$ as a directed connection between node a and node b, the link $[2, 3]$ and the link $[3, 2]$ are two separate entities. The data flow rates leaving each source are denoted by $x_i(t)$, $i \in \mathcal{S}_m$, which we will use as a control input. The total rate of data traversing link j is denoted by $y_j(t)$, $j \in \mathcal{L}_n$. Alternatively, we may denote a link j by its endpoint nodes, $[a, b]$ and the set $\mathcal{L}_n$ as a collection of these pairs. In Fig. 1, we can see that the total data rate on link $y_{[2,3]}$ is equal to the sum of the data rates of $x_{1\mathrm{a}}(t)$ and $x_{2\mathrm{a}}(t)$. Associated with link j is a data rate capacity, denoted by c_j. For the case shown in Fig. 1, if the value of c_j is the same c for all links, and $\frac{c}{2} \leq x_i(t) < c$ for each source, then we can clearly see that the total data rates $y_{[1,4]}, y_{[2,3]}, y_{[4,2]}$, and $y_{[6,5]}$ would all exceed the value of c and that the associated links would represent congested links, and be considered for the congestion control algorithm. All other links would not be congested, and would thus be ignored for this analysis.

Congestion information is returned from the links to the sources through some mechanism and is assumed to be cumulative along the path. As such, in the example in Fig. 1, source $1\mathrm{a}$ receives congestion information that is a sum of the congestion from links $[1, 2]$ and $[2, 3]$.

Given this model, we can describe the dynamics of the queues $q_j(\cdot)$, $j = 1, \ldots, n$, contained at each link in the network as a pure integrator of the excess data rate at the queue with a limiting function given by

$$\dot{q}_j(t) = \begin{cases} \displaystyle\sum_{i\in\mathcal{S}(j)} x_i(t-\tau_i^{\mathrm{f}}) - c_j, & q_j(t) > 0, \\ \max\left(0, \displaystyle\sum_{i\in\mathcal{S}(j)} x_i(t-\tau_i^{\mathrm{f}}) - c_j\right), & \text{otherwise}, \end{cases} \tag{1}$$

with the zero initial conditions $q_j(\theta) = 0,\ \theta \in [-\tau_{\max}, 0]$, $j = 1, \ldots, n$, where $\tau_{\max} = \sup_i \tau_i$. Considering the non-negative initial conditions, we can observe that the value of q_j is restricted to being non-negative, which is consistent with the physical characteristics of a data queue.

4. The Dual Controller

We now introduce the notion of the proportionally fair "dual" congestion control law, which has been mentioned and analyzed extensively [9], [11], [12]. The control inputs $x_i \in \mathbb{R},\ i = 1, \ldots, m$, are the data rates at each source, and the states $p_j \in \mathbb{R},\ j = 1, \ldots, n$, are the "price" at each link such that the feedback control law is given by

$$x_i(t) = x_{\max,i} \exp\left[-\frac{\alpha_i}{g_i \tau_i} r_i(t)\right], \tag{2}$$

where

$$r_i(t) = \sum_{j\in\mathcal{L}(i)} p_j(t-\tau_i^{\mathrm{b}}) \tag{3}$$

and $x_{\max,i} \in \mathbb{R}_+$ is the maximum output data rate for source i, $g_i = |L(i)|$ represents the total number of bottleneck links that the data from source i must traverse, and α_i is a free design parameter that may be set either to a common global value among all sources, or independently at each source. In this model the congestion information is represented by a price $p_j(t)$, given by

$$p_j(t) = \frac{q_j(t)}{c_j}, \quad j = 1, \ldots, n. \tag{4}$$

In order to relate the pricing information and queue dynamics to the congestion control law, we represent the congestion experienced by each of the m sources by a set of m undirected graphs, an example of which appears in Fig. 2. The vertices of each graph represent the congested links in the network. The edges of the graphs represent the subset of links that data from each source traverses. The edges of each graph fully connect the vertices contained in $\mathcal{L}(i)$ for each $i \in \{1, \ldots, m\}$. Additionally, each vertex appearing in $\mathcal{L}(i)$ is self-connected. Each graph can then be represented by an adjacency matrix, A_i, of the form

$$A_{i(k,l)} \triangleq \begin{cases} 1, & k \in \mathcal{L}(i) \wedge l \in \mathcal{L}(i), \\ 0, & \text{otherwise}, \end{cases} \qquad i = \{1, \ldots, m\}, \tag{5}$$

where the notation $M_{(k,l)}$ denotes the kth row and lth column element of the matrix M. Note that by definition $A_i^{\mathrm{T}} = A_i$.

Looking at the example of Fig. 1 and denoting the set of links, $\mathcal{L}_4 = \{[1,4], [4,2], [2,3], [6,5]\}$ as congested (once again assuming that $c_j = c$ for all links and $\frac{c}{2} \leq x_i(t) < c$ for all sources), the set of congestion graphs can be created as in Fig. 2.

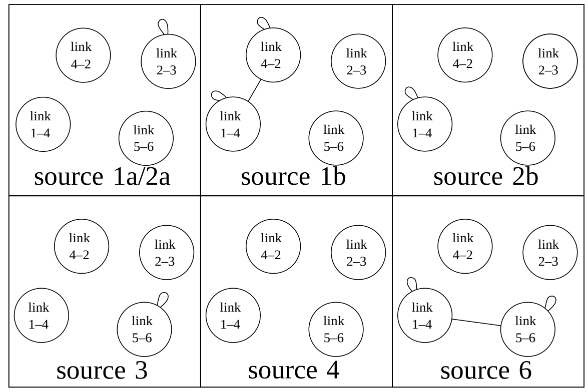

Fig. 2. Congestion graphs of data flow showing the interdependence of the data flow rates resulting from the network topology

5. Stability of the First-Order System

We now assess the stability of the first-order congestion control algorithm. We will make use of the congestion matrices defined in (5) to relate the dynamics of the system that operate on different time scales. We begin by proving an important lemma.

Lemma 1: Let $\mathcal{L}_n$ contain only congested links. Then it follows that $\displaystyle\sum_{i=1}^{m} A_i > 0$.

Proof: We first define a set of vectors $v_i \in \mathbb{R}^n$ such that

$$v_{i(k)} = \begin{cases} 1, & k \in \mathcal{L}(i), \\ 0, & \text{otherwise}, \end{cases} \qquad i = \{1, \ldots, m\}, \tag{6}$$

where the notation $v_{(k)}$ denotes the kth element of the vector v, and note that each A_i of equation (5) can be expressed as $A_i = v_i v_i^{\mathrm{T}} \geq 0$. We also note from the definition (5) that if an off-diagonal element, $A_{i(k,l)}$ has a value of 1, the corresponding diagonal elements, $A_{i(k,k)}$ and $A_{i(l,l)}$ must also have a value of 1. Furthermore, in order to have n congested links, we must have at least $n-1$ connections that traverse only one congested link. If we denote the matrix sum $S = A_1 + A_2 + \ldots + A_m$, then $S_{(k,k)} > S_{(k,l)},\ k \in \{1, \ldots, n\} \backslash a,\ l \in \{1, \ldots, n\},\ l \neq k$, where a is *one* value in the range $1, \ldots, n$. Also $S_{(a,a)} \geq S_{(a,l)},\ l \in \{1, \ldots, n\}$. Simply put, the diagonal elements of the matrix are larger than the off-diagonal elements in every row but one, where the diagonal element is larger than or equal to the off-diagonals. This, combined with the symmetric nature of S, demonstrates that S is full rank since no row can be expressed as a linear combination of any other rows. The fact that S is full rank and non-negative definite concludes the proof. Note that since S is symmetric a similar argument can be made with regard to the columns of S. ■

We now move to the stability analysis of the first-order delay differential system dynamics.

Theorem 2: Consider the network queue dynamical system given by (1) and let $\tilde{p}_j,\ j = 1, \ldots, n$, be defined as the equilibrium prices that satisfy the condition

$$\sum_{i\in\mathcal{S}(k)} x_{\max,i} \exp\left[-\frac{\alpha_i}{g_i \tau_i} \sum_{j\in\mathcal{L}(i)} \tilde{p}_j\right] = c_k \tag{7}$$

at every link k in the system. Under the control law (2), (3) and given the pricing function (4), the solution $p_j(t) \equiv \tilde{p}_j,\ j = 1, \dots, n$, is locally asymptotically stable if $0 < \alpha_i < 1$.

Proof: Equations (1)–(3) and (4) can be combined into a single non-linear differential equation, given by

$$c_j \dot{p}_j(t) = -c_j + \sum_{i \in S(j)} x_{\max,i} \exp\left[-\frac{\alpha_i}{g_i \tau_i} \sum_{k \in L(i)} p_k(t-\tau_i)\right], \tag{8}$$

$p_j(\theta) = 0, \theta \in [-\tau_{\max}, 0]$, that describes the price of a given network link. The linearized link price dynamics around the equilibrium point, $\tilde{p}_j,\ j = 1, \dots, n$, are then defined as

$$\dot{\hat{p}}_j(t) = -\frac{1}{c_j} \sum_{i \in \mathcal{S}(j)} \left[\frac{\alpha_i \tilde{x}_i}{g_i \tau_i} \sum_{k \in \mathcal{L}(i)} \hat{p}_k(t-\tau_i)\right], \tag{9}$$

$\hat{p}_j(\theta) = 0, \theta \in [-\tau_{\max}, 0]$, where

$$\tilde{x}_i \triangleq x_{\max,i} \exp\left[-\sum_{j \in \mathcal{L}(i)} \frac{\alpha_i}{g_i \tau_i} \tilde{p}_j\right], \tag{10}$$

and $\hat{p}_j$ refers to the deviation from the equilibrium such that $\hat{p}_j(t) = p_j(t) - \tilde{p}_j$.

Using the adjacency matrix A_i from (5), (9) can be expressed as a matrix equation describing the price of sending data on each link in the network

$$\dot{\hat{p}}(t) = C^{-1} \sum_{i=1}^{m} \frac{\alpha_i \tilde{x}_i}{g_i \tau_i} A_i \hat{p}(t-\tau_i) \tag{11}$$

where $C \in \mathbb{R}^{n \times n}$ is a diagonal matrix having the values of $-c_j$ on the diagonal and $\hat{p} \triangleq [\hat{p}_1, \dots, \hat{p}_n]^{\mathrm{T}}$. Equation (11) is referred to as a retarded differential difference equation [13] with multiple non-zero delays, where α_i is a free-design parameter.

From (4) we note that the queue length varies linearly with respect to the queue price. As such, the asymptotic stability of the queue price is sufficient to imply the asymptotic stability of the queue length and the entire network system. Therefore, we will prove the asymptotic stability of the linearized system given by (11).

We note that since the function $p(t)$ is continuous for $t > 0$, the time delay expression can be expressed as

$$\hat{p}(t) = \hat{p}(t-\tau) + \int_{-\tau}^{0} \dot{\hat{p}}(t+s)\,\mathrm{d}s. \tag{12}$$

We now define a Lyapunov-Krasovskii functional candidate given by

$$\begin{aligned} V(\hat{p}) &= V_1(\hat{p}) + V_2(\hat{p}), \\ V_1(\hat{p}) &\triangleq \hat{p}^{\mathrm{T}} \left[\sum_{i=1}^{m} \frac{\alpha_i \tilde{x}_i}{g_i \tau_i} A_i\right] \hat{p}, \\ V_2(\hat{p}) &\triangleq \sum_{i=1}^{m} \frac{\alpha_i \tilde{x}_i}{g_i \tau_i} \int_{-\tau_i}^{0} \int_{t+s}^{t} \dot{\hat{p}}^{\mathrm{T}}(\theta) A_i \dot{\hat{p}}(\theta) \mathrm{d}\theta\,\mathrm{d}s, \end{aligned} \tag{13}$$

and expand the derivative of the two terms separately for simplicity. We first note that the matrix in V_1 is indeed positive definite. From Lemma 1, the sum of all A_i defined in (5) is positive definite. Consequently, since $\tilde{x}_i, \tau_i$, and g_i are all positive, the sum

$$M = \sum_{i=1}^{m} \frac{\alpha_i \tilde{x}_i}{g_i \tau_i} A_i, \tag{14}$$

where $M \in \mathbb{R}^{n \times n}$ is positive definite for any $\alpha_i > 0$. A similar argument holds for V_2.

Differentiating the expression for V_1 and utilizing (12) yields

$$\begin{aligned} \dot{V}_1(\hat{p}(t)) &= 2\dot{\hat{p}}^{\mathrm{T}}(t) \sum_{i=1}^{m} \frac{\alpha_i \tilde{x}_i}{g_i \tau_i} A_i \hat{p}(t) \\ &= 2\dot{\hat{p}}^{\mathrm{T}}(t) C \left[C^{-1} \sum_{i=1}^{m} \frac{\alpha_i \tilde{x}_i}{g_i \tau_i} A_i \hat{p}(t-\tau)\right] \\ &\quad + 2\dot{\hat{p}}^{\mathrm{T}}(t) \sum_{i=1}^{m} \frac{\alpha_i \tilde{x}_i}{g_i \tau_i} A_i \int_{-\tau_i}^{0} \dot{\hat{p}}(t+s)\,\mathrm{d}s, \end{aligned} \tag{15}$$

where the term in the square brackets can then easily be seen to be the same as (11), allowing (15) to be rewritten as

$$\begin{aligned} \dot{V}_1(\hat{p}(t)) &= 2\dot{\hat{p}}^{\mathrm{T}}(t) C \dot{p}(t) \\ &\quad + 2\sum_{i=1}^{m} \frac{\alpha_i \tilde{x}_i}{g_i \tau_i} \int_{-\tau_i}^{0} \dot{p}^{\mathrm{T}}(t) A_i \dot{p}(s+t)\,\mathrm{d}s \end{aligned} \tag{16}$$

after rearranging the scalar terms in the equation. We note that each A_i is a non-negative definite matrix, and can be expressed as the product $A_i^{1/2} A_i^{1/2}$, where $(\cdot)^{1/2}$ denotes the symmetric non-negative square root of a matrix. Combining this with the inequality

$$2a^{\mathrm{T}} b \leq a^{\mathrm{T}} a + b^{\mathrm{T}} b \tag{17}$$

for any vectors a and b, and taking $a = A_i^{1/2} \dot{\hat{p}}(t)$ and $b = A_i^{1/2} \dot{\hat{p}}(s+t)$, the expression of (16) can be restated as

$$\begin{aligned} \dot{V}_1(\hat{p}(t)) &\leq 2\dot{\hat{p}}^{\mathrm{T}}(t) C \dot{\hat{p}}(t) + \sum_{i=1}^{m} \frac{\alpha_i \tilde{x}_i}{g_i} \dot{\hat{p}}^{\mathrm{T}}(t) A_i \dot{\hat{p}}(t) \\ &\quad + \sum_{i=1}^{m} \frac{\alpha_i \tilde{x}_i}{g_i \tau_i} \int_{-\tau_i}^{0} \dot{\hat{p}}^{\mathrm{T}}(t+s) A_i \dot{\hat{p}}(s+t)\,\mathrm{d}s. \end{aligned} \tag{18}$$

Turning to the expression for V_2 and differentiating yields

$$\begin{aligned} \dot{V}_2(\hat{p}(t)) = \sum_{i=1}^{m} \frac{\alpha_i \tilde{x}_i}{g_i \tau_i} \int_{-\tau_i}^{0} \Big[&\dot{\hat{p}}^{\mathrm{T}}(t) A_i \dot{\hat{p}}(t) \\ &- \dot{\hat{p}}^{\mathrm{T}}(t+s) A_i \dot{\hat{p}}(t+s)\Big]\,\mathrm{d}s. \end{aligned} \tag{19}$$

Summing (18) and (19) then gives the expression

$$\dot{V}(\hat{p}(t)) \leq 2\dot{\hat{p}}^{\mathrm{T}}(t) C \dot{\hat{p}}(t) + 2\sum_{i=1}^{m} \frac{\alpha_i \tilde{x}_i}{g_i} \dot{\hat{p}}^{\mathrm{T}}(t) A_i \dot{\hat{p}}(t). \tag{20}$$

Factoring then yields

$$\dot{V}(\hat{p}(t)) \leq 2\dot{\hat{p}}^{\mathrm{T}}(t) Q \dot{\hat{p}}(t), \tag{21}$$

where

$$Q = \left[C + \sum_{i=1}^{m} \frac{\alpha_i \tilde{x}_i}{g_i} A_i\right]. \tag{22}$$

Thus, $\dot{V}(\hat{p}(t)) \leq 0$ for all $\dot{\hat{p}} \neq 0$ if Q is a negative definite matrix. We now derive the condition on α_i that guarantees the negative definiteness of Q.

Let us examine the case where $\alpha_i = 1,\ i \in \{1,\ldots,m\}$, and define $\bar{Q}$ as the value of Q when all α_i are unity. Then we define the function $(\cdot)_r$ of a matrix as the sum of the elements of the rth row of the matrix. If we then examine the quantity

$$\left(\bar{Q}\right)_r = (C)_r + \sum_{i=1}^{m} \frac{\tilde{x}_i}{g_i} (A_i)_r\,, \quad r \in \{1,\ldots,n\}, \tag{23}$$

we can note that the quantity $(C)_r = -c_r$, the link capacity of link r, and that from (5) $(A_i)_r = |\mathcal{L}(i)| = g_i$ if $i \in \mathcal{S}(r)$ or 0 otherwise. As such, the row summation of (23) becomes

$$\left(\bar{Q}\right)_r = \left(-c_r + \sum_{i\in\mathcal{S}(r)} \tilde{x}_i\right) = 0, \quad r \in \{1,\ldots,n\}, \tag{24}$$

from (7). We then can say that $\bar{Q}$ has the following properties:

(i) the sum of each row (column) is 0.
(ii) the matrix is symmetric since it is the sum of symmetric matrices
(iii) the diagonal and only the diagonal consists of negative elements,

which is sufficient to prove the non-positive definiteness of $\bar{Q}$ by expanding the quadratic form $x^{\mathrm{T}}\bar{Q}x$.

From simple matrix operations it follows that

$$x^{\mathrm{T}}\bar{Q}x = \sum_{k=1}^{n} x_{(k)} \left(\sum_{l=1}^{n} \bar{Q}_{(k,l)} x_{(l)}\right), \tag{25}$$

where the subscripts represent the row and column indices of the associated vectors and matrix. From the properties of $\bar{Q}$,

$$\bar{Q}_{(k,k)} = -\sum_{l=1;l\neq k}^{n} \bar{Q}_{(k,l)}, \tag{26}$$

leading to

$$\begin{aligned} x^{\mathrm{T}}\bar{Q}x &= \sum_{k=1}^{n} x_{(k)} \left(\bar{Q}_{(k,k)} x_{(k)} + \sum_{l=1;l\neq k}^{n} \bar{Q}_{(k,l)} x_{(l)}\right) \\ &= \sum_{k=1}^{n} x_{(k)} \left(\sum_{l=1}^{n} \bar{Q}_{(k,l)} (x_{(l)} - x_{(k)})\right) \\ &= -\sum_{k=1}^{n} \left(\sum_{l=1}^{n} \bar{Q}_{(k,l)} (x_{(k)}^2 - x_{(k)} x_{(l)})\right). \end{aligned} \tag{27}$$

Also, since $\bar{Q}$ is symmetric $\bar{Q}_{(k,l)} = \bar{Q}_{(l,k)}$ and the associated terms from the summation in (27) can be added together, yielding

$$\begin{aligned} x^{\mathrm{T}}\bar{Q}x &= -\sum_{k=1}^{n} \left(\sum_{l=k}^{n} \bar{Q}_{(k,l)} (x_{(k)}^2 - 2x_{(k)}x_{(l)} + x_{(l)}^2)\right) \\ &= -\sum_{k=1}^{n} \left(\sum_{l=k}^{n} \bar{Q}_{(k,l)} (x_{(k)} - x_{(l)})^2\right) \leq 0, \end{aligned} \tag{28}$$

since all of the off-diagonal elements are positive. Now, we define $Q|_{\alpha_i<1}$ as being the value of the matrix Q when $\alpha_i < 1$. In can then be seen that $Q|_{\alpha_i<1} < \bar{Q},\ i \in \{1,\ldots,m\}$, since each A_i is non-negative definite and C is negative definite. We can then conclude that

$$Q|_{\alpha_i<1} < 0, \quad i \in \{1,\ldots,m\}. \tag{29}$$

We can then further state that from (21) the system described by the dynamics in (11) is Lyapunov stable if the value of $\alpha_i < 1,\ i \in \{1,\ldots,m\}$.

It can be noted that the derivative of the Lyapunov-Krasovskii functional along the system trajectories is 0 if and only if $\dot{\hat{p}}(t) = 0$. Since $\hat{p}(t)$ is bounded, $\hat{p}(t) = \hat{p}_{\mathrm{c}}$, a constant, when $\dot{V}(\hat{p}(t)) = 0$. Given this, the system dynamics at that time are given by

$$\dot{\hat{p}}(t) = \left[C^{-1} \sum_{i=1}^{m} \frac{\alpha_i \tilde{x}_i}{g_i \tau_i} A_i\right] \hat{p}_{\mathrm{c}}. \tag{30}$$

Thus, since the square bracketed matrix has full rank (from Lemma 1), and since $\dot{\hat{p}}(t) = 0$, it follows that $\hat{p}(t) \equiv \hat{p}_{\mathrm{c}} \equiv 0$ and the system is asymptotically stable about the equilibrium point. ■

6. Second-Order Dynamics

While the first-order system of (11) is stable, it does not allow us to directly control the behaviour of the queues in the network. If for example, we wished to maintain the steady-state queue level within some range or about some value, we have no mechanism to do this. As such, we propose a system with higher order dynamics, similar to the dynamics of the XCP protocol [2]. We modify the function $r_i(t)$ from (3) to be

$$r_i(t) = \sum_{j\in\mathcal{L}(i)} p_j(t - \tau_i^{\mathrm{b}}) + \frac{\beta_i}{\alpha_i \tau_i} \int_0^t p_j(s - \tau_i^{\mathrm{b}}) - p_j^0 \,\mathrm{d}s, \tag{31}$$

where β_i, like α_i, is a user definable tuning parameter, and p_j^0 represents the link cost at the desired equilibrium queue depth. This effectively specifies a PI controller allowing us to control the steady state queue depths. To simplify the notation, we assume that the values of $p_j^0 = 0,\ j = 1, 2, \ldots, n$.

Similar to (8) we can express the complete non-linear dynamics of this system as a delay-differential equation only in $p_j(t),\ j = 1, 2, \ldots, n$ as

$$\begin{aligned} c_j \dot{p}_j(t) = -c_j + \sum_{i\in S(j)} x_{\max,i} \\ \cdot \exp\left[-\frac{1}{g_i\tau_i} \sum_{k\in L(i)} \alpha_i p_k(t-\tau_i) + \frac{\beta_i}{\tau_i}\int_0^t p_k(s-\tau_i)\,\mathrm{d}s\right], \end{aligned} \tag{32}$$

which can be linearized for analysis as

$$\dot{\hat{p}}(t) = C^{-1} \sum_{i=1}^{m} \frac{\alpha_i \tilde{x}_i}{g_i \tau_i} A_i \hat{p}(t-\tau_i) + C^{-1} \sum_{i=1}^{m} \frac{\beta_i \tilde{x}_i}{g_i \tau_i^2} A_i \int_0^t \hat{p}(s-\tau_i)\,\mathrm{d}s, \tag{33}$$

where $\tilde{x}_j$ is redefined as

$$\tilde{x}_i \triangleq x_{\max,i} \exp\left[-\sum_{j\in\mathcal{L}(i)} \frac{\alpha_i}{g_i\tau_i}\tilde{p}_j + \frac{\beta_i}{\tau_i}\tilde{p}_j^{\mathrm{int}}\right], \tag{34}$$